Commodore INFO

Snelle disk-drive

Homeword

Reken LOGO

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS PRIJS ƒ 7,00/Bfr. 130 JAARGANG 3, No. 2, 1986

LISTINGS



DIVERSE AANBIEDINGEN MET KORTINGS BONNEN!

Misdaad op de 64

Sprites of mobs

Adventures

Rij-examen per computer

Controleer uw software

Miniatuurtjes

Machinetaal deel 9
Basic cursus deel 3

Redactioneel

Na een winterseizoen, waarin Commodore haar plaats als leidende producent van huiscomputers nauwelijks in gevaar gebracht zag, komt de lente nu met veel actie. Voor de groep veeleisende gebruikers is de Amiga nu ook officieel hier.
Daarmee heeft Commodore weer een nieuw en zeer heet ijzer in het vuur. Een machine met veel nieuwe mogelijkheden, die zowel voor recreatieve toepassingen, dus spelletjes, als zakelijk een nieuwe horizon ontsluiten. Een machine, waarvan de prijsstelling tot het laatste moment geheim gehouden wordt, maar die in een complete uitvoering toch wel boven de 6 mille komt. Gelukkig komen er ook basismodellen, waardoor men geleidelijk kan uitbouwen met monitor en diskdrives. In ieder geval een machine, die voor de voortrekkers in computerland een must is.
Een en ander betekent niet, dat we de 64/128 en C16 bezitters vergeten, voor hen blijft dit blad natuurlijk in de eerste plaats. Maar u heeft al gemerkt, dat we de Amiga en de PC een eigen plaatsje gunnen en misschien binnenkort met een apart blad de bezitters daarvan gaan bedienen.

L.Sala

Inhoud van dit nummer

PC-TAKS

Er blijkt in de PC wereld grote behoefte aan een relatief goedkoop belastingpakket. Infolist is nu bezig om het Taks-64 pakket om te laten zetten naar MS-DOS formaat. PC-TAKS gaat ƒ 99,- incl. BTW. kosten. Geinteresseerden kunnen kontakt opnemen via 02152-62343, of gebruik maken van de bestelbon elders in het blad.

COMMODORE-INFO
verschijnt 10× per jaar
Jaargang 3, no. 2, 1986
Uitgave:
Sala Communications
Uitgever:
Drs. J. Taverne
REDAKTIE
Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdred.
K. van der Vlies reportage
R. van den Heuvel listings
B. Munniksma onderwijs
Sj. Bakker machinetaal
ART DEPARTMENT
Strip:
Bert Tier
Illustraties:
Ben van Mierlo
Ymmot
Advertentie-exploitatie:
Ing. V. Sala
Redaktieadres:
Postbus 112
1260 AC Blaricum
☎ 02152-65695
Abonnementen en administratie:
Postbus 5570
1007 AN Amsterdam
Den Texstraat 5a
1017 XW Amsterdam
☎ 020-273198
Vragen betreffende abonnementen bij voorkeur schriftelijk, met meesturen van het omslagetiket. Telefonisch uitsluitend donderdag en vrijdag van 10.00 tot 15.00 uur.
Abonnement:
f 55,– of Bfr. 1100 per jaar
(10 nummers)
Betaling op Giro 1585491 tnv. SAC/COMMODORE-INFO Blaricum of in België op Bank BBL nr. 310050602562, vermeld SAC/COMMODORE-INFO. Oude nummers à *f* 6,75 alleen bij vooruitbetaling op giro 1585491.
Redactiesecretariaat:
F. Belmer
J. Emmelot
Druk:
Verweij, Mijdrecht
NDB, Zoeterwoude
Distributie:
in Nederland Betapress/Gilze
in België AMP/Brussel

ISSN: 0169-3085

nieuws nieuws nieuws

Game Killer

Heeft u ook van die videospelen waarbij je nooit door al die schermen heen komt? Lijkt er eindelijk enige vooruitgang geboekt dan volgt toch weer, enkele spelschermen verder, de onvermijdelijke spritecrash. Ieder van ons voelt wel eens de behoefte om stiekem eens door het videogame te lopen en de moeilijkheden te analyseren. De Sprite Killer van Micro City biedt daartoe ruimschoots de gelegenheid.

Sprite Killer bestaat uit een nette zwarte cartridge met aan de zijkant een geel knopje. Steek de cartridge in het slot op uw C64 of C128 en start de machine op. Bij de C128 moet de gebruiker tijdens het opstarten de CBM-toets ingedrukt houden!

Het tv/monitorscherm toont vervolgens wat laadvuurwerk en na enige seconden verschijnen de attractieve graphics met semi 3D- effecten, wandelende letters en lichtkrant. Een pittig melodietje brengt de stemming er direkt in. U kunt de hele lichtkrant uitlezen of gewoon direkt op de spatiebalk drukken waarna het CBM-scherm met READY verschijnt. Het laden van cassette of disk verloopt verder zoals gebruikelijk.

Game Killer biedt de volgende gebruikersopties:

❶ Het uitschakelen van alle spritebotsingen; druk de gele knop kort in. Bij lang indrukken valt het beeld uit.

❷ Het uitschakelen van onderlinge spritebotsingen; druk de joystick naar voren en de gele knop in.

❸ Het uitschakelen van sprite-achtergrond/scenerybotsingen; druk de joystick naar achteren en de gele knop in.

Het eerste spel dat wij probeerden, Star Race van Mastertronic, liet zich mooi niet killen. Nadere informatie bij Micro City leerde ons dat de Game Killer alleen bij machinetaal programma's werkt. Ruwweg dus bij 75% van alle spelen en niet bij de overige, meestal uit oude BASIC-games bestaande, 25%.

Een tweede game liet zich met gemak doorlopen. Zonder een enkele klapper bewoog de spelsprite moeiteloos van scherm naar scherm en nam elk obstakel zonder mankeren. Daarbij was het goed mogelijk om alle opgeworpen problemen te analyseren.

Eindelijk was het mogelijk om de 500 Eindeloosschermen van Radarsoft ongestraft te doorlopen. Een ware opluchting.

Game Killer biedt de gefrustreerde speler de mogelijkheid om alle botsingsproblemen tussen de sprites onderling en met de spelscenery haarfijn uit te pluizen. Bij tekstadventures e.d. werkt de Sprite Killer uiteraard niet. Of het inzetten van spritemoordenaars wel met de sportiviteit strookt blijft een kwestie van opvatting.

Koronis Rift

Koronis Rift is het nieuwste videospel van LucasFilm-games. Deze maal is de speler een techno-scavenger, een



soort Science Fiction-sloper die het heelal afstroopt naar verloren gegane technologiën. Aan het begin is het u eindelijk gelukt om het fabelachtige Koronis Rift te vinden. Koronis Rift was in een ver verleden de technologische proeftuin van de mytische ouden. De speler rijdt in een Surface Rover over het planeetoppervlak. Het scherm laat het uitzicht van de bestuurder op diens omgeving zien en biedt verder, zoals we dat van LucasFilm-games gewend zijn, een schat aan indicatorinformatie. De ontwerpers hebben er weer van alles aan gedaan om de fictieve realiteit zo dicht mogelijk te benaderen.

Het spel zelf is voornamelijk van strategische aard. Al die technologische rijkdom wordt namelijk vol ferve bewaakt door een aantal niets ontziende genetisch gemanipuleerde mutanten. Deze Guardians leggen de speler het vuur danig aan de schenen en zonder een juiste wapenkeuze en zorgvuldige taktiek is het spel dan ook snel afgelopen. De enige steun komt van het ScienceDroid-hulpje van de ruimtesloper.

Aanbevolen voor de liefhebbers van goed verzorgde taktische schietspelen.

Lode Runner

In Japan is Broderbund's Lode Runner een echte knaller en ook in de V.S. begint dit spel zijn intree in de gamehalls te maken. Inmiddels zijn er al diverse Lode Runner versies in omloop. Op elk scherm dient de athletische spelheld kostbaarheden te vergaren en daarbij uit de handen der bewakerrobots (Bungeling Guards) te blijven. De scenery bestaat uit ladders, touwen en stenen doolhofstructuren. In de originele Lode Runner-versie kan met behulp van de screen editor zelf een van de 50 schermen gekozen worden. Bij **Championship Lode Runner** moet echter elk scherm uitgespeeld worden en de voltooiing van dit videogame neemt dan ook wel enige maanden in beslag. Wel bestaat de mogelijkheid om de gewonnen schermen te saven. Typisch Amerikaans is het codewoord, te verkrijgen door ook het laatste spelscherm te nemen, dat bij Broderbund voor een kampioenscertificaat ingewisseld kan worden. Bovendien is ter vergemakkelijking ook nog een hintboek voor *f* 30,- leverbaar.

Nieuw is **Lode Runners Rescue**. Een 3D-versie die aan het klassieke arcadespel Crystal Castles doet denken. Dit-

maal dient de dochter van Lode Runner haar vader uit de handen van de Bungeling Guards te redden. Daarvoor dienen over 46 schermen verspreide spelsleutels veroverd te worden.
Kenmerkend voor de spelscenery zijn de vierkantjes die als lift, water, kussens, vallen, krachtpaddestoel e.d. functionren. Een Restart Goal maakt het mogelijk om bij het verlies van een leven naar de uitgangspositie i.p.v. helemaal naar het begin terug te keren. Ook kan het bereikte schermniveau gesaved worden.
Verder maakt een screen editor het wijzigen of verplaatsen van schermen en schermonderdelen mogelijk.
Lode Runner-keuze genoeg lijkt ons zo.

Bank Street Mailer

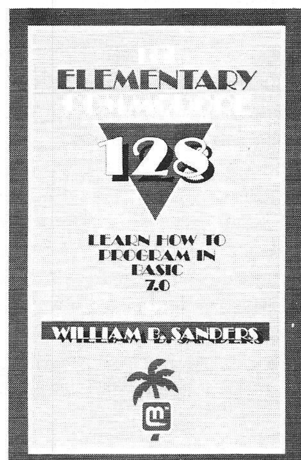
In de reeks Bank pakketten van Broderbund is nu ook het mailing lijst pakket beschikbaar. Voor *f* 139,- via Aackosoft is dit een nuttige aanvulling voor een file-management programma. Er zit bij dit pakket ook een tekstverwerker om korte brieven etc. te kunnen maken en met persoonlijke gegevens aan te vullen.

Spelenderwijs programmeren op de 64

Van uitgeverij Luitingh ontvingen wij een nieuw programmeerboek voor de 64. Het richt zich vrijwel helemaal op het genre "Adventure" en men vult de 240 bladzijden nogal snel met grote letters en erg veel wit. Het hart van het boek vormt het adventure "ondergronds avontuur". Van een pril basis-idee wordt men spelenderwijs via een scenario, opzet, kaarten maken, voorwerpen benoemen en subroutines maken tot een compleet programma gebracht. Wat simpeltjes, maar leuk om te lezen. Luitingh, *f* 29,90

The Elementary C-128

Dit Engelstalige boek van William Sanders belooft het succes van zijn gelijknamige werk voor de 64 te evenaren. Met 250 pagina's gedegen informatie, tamelijk dicht op elkaar, is dit boek voor de beginnende 128-er, maar ook voor ervaren 64-doorgroeiers aan te bevelen. De materie wordt tamelijk ver uitgediept, zo staat er bij Sorteren oa. Bubble Sort, Shell Sort en Quick Sort. Het boek geeft voor vrijwel alle onderwerpen niet alleen een voorbeeld, maar dat is dan meestal tegelijk een handige subroutine. Bruikbare stringbewerkingen, inputroutines, wie dit boek heeft doorgewerkt, kan er tegen aan. Het betere 128 boek! Uitgeverij Micromscribe *f* 44,90.



C-64 Handleiding

Calimero, bekend als verkopers van goedkope 64's, had behoefte aan een nederlandse handleiding bij hun produkten. Die maakten ze, en het resultaat is á *f* 22,50 ook te koop (010- 155716.

Het zwaardere werk op de 64

Weka is al bekend als leverancier van zeer lijvige en gedegen naslagbanden voor de computeraar. Wie echt alles over computerspelen wil weten, kan voor *f* 99,- echt honderden pagina's spelinformatie in handen krijgen. Erg up to date met speltips voor Way of the Explo-

Gebruikersgroepen

De **VCGN** (070-950779) is de oudste Commodore gebruikersvereninging en heeft iedere laatste zaterdag van de maand in het Lodewijk Makeblijde lyceum, Haagweg 155 in Rijswijk haar grote bijeenkomst.
Inlichtingen over de vele afdelingen kunt u krijgen via:
Zeeland 01180-25746 West-Brabant 01640-42404 Midden-Brabant 01612-54242 Oost-Brabant 04120-42054 Zuid-Limburg 04490-19946 Zuid-Holland 078-135739

PET Benelux heeft bijeenkomsten in Zeist (1e zaterdag), Nijmegen (2e zaterdag) en Haaksbergen (3e zaterdag). (Inl.05759-2211).

De **HCC** Commodore gebruikersgroep heet Compet, 03404-59599.

HOT NEWS van SCN uit Amsterdam, langzamerhand de grootste vereniging van ons land, (03450-16051), heeft haar maandelijkse bijeenkomst in de Meervaart op de 2e zaterdag van de maand en ook activiteiten in Zwolle (038-548459), Midden Nederland (03453-1375) en Deventer (05700-50758).

CCCB is één van de clubs uit Belgie, met afdelingen in Antwerpen, Limburg, West-Vlaanderen, inl. Bindstr. 19, te 2600 Berchem Antwerpen. Lid worden door storting 500 fr op 405-5092171-90 tnv. CCCBA.

In **Arnhem** zit een Commodore-club, waar men iedere twee weken op dinsdag clubavonden organiseert en ook cursussen geeft. Inl. Hofsingel 259 te 6834 GH Velp, tel. 085-647782.

In **Noordwijk** zit een C-64/128 gebruikersgroep, die iedere 1e donderdag een bijeenkomst heeft. Inlichtingen A. vd Klaauw ☎ 01719-16720.

In **Knokke** is er iedere vrijdag een bijeenkomst in Café Montyu.

ding Fist, Boulder dash II etc. etc. Jaren spelervaring nu vastgelegd en nog handig na te zoeken ook.

Hobbyscoop

Wekelijks op woensdag (19.00 uur) wordt de BasiCode beeldkrant uitgezonden. Die beeldkrant bevat ongeveer 8 vellen informatie over de in Hobbyscoop besproken onderwerpen en een agenda over computer-events.

Supertoolkit

Alweer een nieuwe module, deze keer van Cat & Korsh uit Rotterdam. Zij brengen voor ƒ 69,- een insteek-cartridge met erg veel mogelijkheden, zoals tape- en disk turbo, 50 extra commando's en uitgebreide copy-faciliteiten. Oook komt uit dit huis de uit Duitsland stammende **SPEEDDOS PLUS 1**, in de '86 versie. Dit is een naar wens in te schakelen diskversneller in modulevorm, die voor ƒ 179,- te koop is. Er zit dan ook nog een uitbreiding voor 2x 8KB extra ROM in, voor eigen of andere EPROM firmware. (inl. 010-180416).

Robotkit

De Robot Division van **Spectron Instrument** heeft voor de VIC 20 en C64 een veelzijdige Robotkit ontwikkeld. De Micro-Kitten kit beschikt over een breed toepassingsgebied: Tekenen, transport, het bouwen van geautomatiseerde modellen,spelletjes en educatie. Als toebehoren zijn diverse sensoren, radiografische besturing en zgn. **Brains on Board** verkrijgbaar.
De kit bevat naast de Micro-Kitten zelf een aantal beginnersprogramma's, instructies, computerinterface en onderdelen voor de verschillende robottaken. Het bouwpakket kost rond de ƒ 250,- en de gemonteerde versie rond de ƒ 360,-.Verdere info bij Spectron Instrument, Robot Division, MS 24, 1342W. Cedar Avenue, Denver, CO 80223, USA.

Emmy Award voor Time Processor C64

Voor het eerst in de geschiedenis werd een Emmy Award uitgereikt voor een softwarepakket. Deze fel begeerde filmprijs ging naar Richard en Ron Grant van Sherman Oaks, Californië voor hun pakket Time Processor. Time Processor stelt de C64(128)- gebruiker in staat soundtracks voor tv- of filmscripts nauwkeurig op de vertoonde handelingen te synchroniseren. Via de schermdisplay kan de componist het tempo en de maten van de filmmuziek optimaal timen. Een ingebouwde hoorbare metronoom vergemakkelijkt het dirigeren.
De Time Processor werd ondermeer met succes gebruikt bij bekende tv-series als Dallas, Knight Rider en voor de muzikale omlijsting van de Olympische spelen.

Twister

Een demonisch 3D spel dat u vele slapeloze nachten zal bezorgen. Twister is de vrouwelijke personalisatie van duivelse krachten die de speler dient te vernietigen. Twister weet zich verzekerd van de hulp van geesten, demonen, duivels en monsters die u in het 3D mazennetwerk zult moeten elimineren. Dit mazennetwerk lijkt veel op de lijnengraphics zoals wij die uit de 3D-spelen Elite en Mercenary kennen. De animatie verloopt vrijwel perfect en de gebruikte graphics slepen alles uit de C64 wat er in zit.
Het vernietigen van de demonische hulptroepen en de kwade Twister zelf vergt veel strategisch inzicht. Ook in deze volgt Twister de met Elite ingeslagen weg.
Van het Britse System 3 voor ongeveer dertig gulden. ●

Boekenrubriek

40 Grafische Programma's voor de C-64

M.Sutter, *ƒ* 29.50

Dit boek, met als subtitel **Leer programmeren met hoge resolutie graphics in Basic**, geeft een serie listings, waarbij met hulp van de Hires-mode van de C-64 de nodige al of niet wiskundige figuren op je scherm worden getoverd. Hetzelfde boek is onlangs verschenen voor de MSX-computers, en dat blijkt ook uit de in het boek gebruikte Basic-syntax. Er wordt vanuit gegaan, dat de C-64 gebruiker beschikt over **Simons Basic**, zodat de commando's HIRES en LINE kunnen worden toegepast. Voor degenen die niet over deze utility beschikken is achter in het boek de Basic-versie van deze commando's opgenomen, maar dat betekent dat je bij het runnen wel een paar minuten geduld moet hebben, voordat er ook maar iets op je grafische scherm verschijnt.
De 40 listings zijn over het algemeen niet langer dan zo'n 50 regels Basic, dus vrij snel in te toetsen. Het eerste deel behandelt de meer wiskundige figuren, zoals drie-dimensionale driehoeken, moiree-effecten en ingeschreven vierkanten.
Verder krijg je sinuskrommen, parabolen en andere cartesische vormen voorgeschoteld. Histogrammen en geografie zijn zelfs niet vergeten, en bij iedere toepassing vind je één of meer illustraties van het grafische scherm, zodat je voordat je aan een programma begint alvast kunt bekijken hoe de figuren eruit gaan zien.
Als appendix zijn een paar (grote) machinetaal-programma's opgenomen, die zijn overgenomen uit het Amerikaanse tijdschrift Commodore Computing, waarmee de nodige Hires-opdrachten door machinetaal kunnen worden vervangen.
Voor degenen onder ons, die wat meer willen doen met de grafische mogelijkheden van de 64, een goed boek om mee van start te gaan. Wel is het aan te bevelen, om hierbij toch Simons Basic te gebruiken. Beschik je over een meer dan gemiddelde kennis van de 6510 machinetaal, dan zijn de assembly-listings achter in het boek duidelijk genoeg om zelf erg snel in Hires te kunnen gaan werken.
Voor de prijs de moeite waard.

Werken met de Commodore 64

Peters, SMD

Een "originele" titel, bedacht door de heren P.Peters en H.Peters, de auteurs van dit boek. Uitgegeven door SMD, Leiden. De eerste 100 pagina's van dit werk worden gewijd aan de beschikbare Basic-opdrachten en -commando's. Naast een korte uitleg, bedoeld voor de lezers op nulnivo, krijg je voor ieder commando een paar voorbeelden en opgaven. Deze zijn niet altijd even duidelijk en goed gekozen. Complete listings, die meerdere opdrachten verenigen tot een nuttig programma zul je hierin vergeefs zoeken.
Het tweede deel van dit boek geeft een nederlandse handleiding voor standaard software als de tekstverwerker EasyScript, C-64 BASE bestandsverwerking en het rekenmatrix/ calculatieprogramma ABRA-CALC. Heb je (toevallig) één van deze pakketten en ben je de originele gebruiksaanwijzing kwijt, dan kan dit boek uitkomst bieden.
Het boek zou goed kunnen dienen als leerboek voor een computer- cursus, waarbij de nodige persoonlijke begeleiding het geheel wat kan verduidelijken. Bezit je het C-64 handboek en dito boeken voor je software, of weet je al het één en ander van Basic, dan kun je dit boek rustig in de winkel laten liggen.

Doe het zelf programma's op de COMMODORE-64

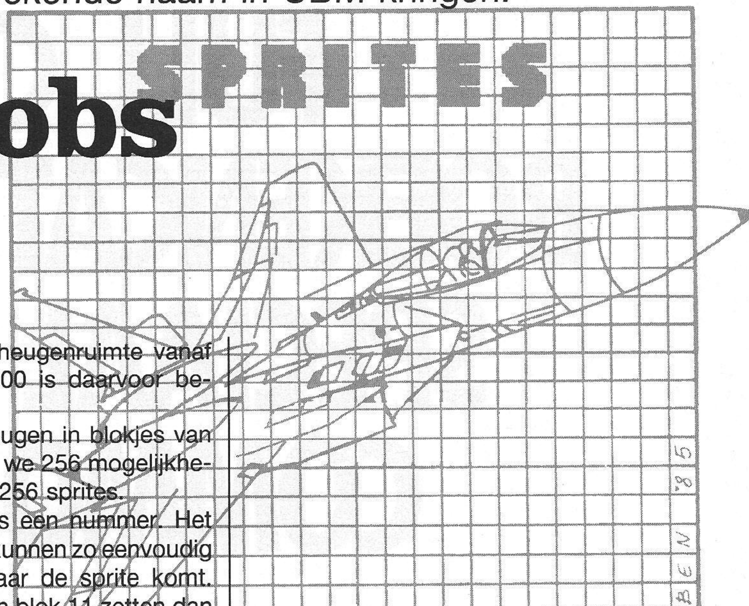
D.- Kreutner *ƒ* 39,90

Dit door Academic Service uitgegeven boek ligt sinds mei 1985 in de boekhandel. In tegenstelling tot de meeste andere boeken over dit onderwerp zitten er in dit boek nogal wat nuttige listings, waarmee je alle kanten uitkunt, ook zonder het naadje van de Basic-kous te weten. Niet alles is even origineel, maar dat kan ook niet voor 40 gulden. De meeste programma's zijn voorzien van een duidelijke uitleg, waar het de kernroutines betreft, zodat het niet zo moeilijk wordt om de programma's aan te passen aan de eigen behoefte.
Het boek is bedoeld voor de C-64 bezitters, die werken met een cassette-recorder, maar de meeste listings zijn voorbereid om over te zetten naar gebruik van een diskdrive en (waar nodig) de printer. Verwacht geen sterke staaltjes van flitsende machinetaal, het is alles puur en rudimentair Basic wat de klok slaat. Wat te denken van het programma PIEP1 waarbij je steeds een boodschap op je scherm print, die vergezeld wordt van een aantal door de gebruiker in te geven piepjes, (PIEP2 is er ook natuurlijk), of KRIJGERTJE, waarbij je in slow-motion een arcadespel kunt spelen.
De **zakelijke** toepassingen hebben wat meer aandacht gekregen, en de serieuze beginners kunnen aan de hand van dit boek vrij goed het werken met sequentiële bestanden onder de knie krijgen. De relatieve bestanden worden hierbij (helaas) overgeslagen.
Het boek eindigt met enkele Basic-programma's, die samen de opzet vormen voor een (simpele) boekhouding.
Hoewel in Nederland in 1985 uitgegeven, vinden we de stand van zaken terug in dit boek, zoals die in 1983 bestond, rond de C-64. Er wordt als randapparatuur b.v. nog verwezen naar de printers die voor de VIC-20 bestemd waren en het ook wel deden op de 64 en zo nog wel wat zaken, maar de Basic-syntax is in de jaren niet veranderd, dus op zich hoeft dat geen beletsel te zijn, maar toch.....

J.B. ●

Buiten karakters/tekensets en bit-map-grafiek zijn sprites de derde mogelijkheid om iets grafisch zichtbaar te maken op de Commodore 64. Ze kunnen net als karakters worden gebruikt om in spelletjes 'leuke' tekens te maken voor b.v. ruimteschepen. Als "bit-map- graphics" kunnen ze ook voor kleine grafische plaatjes gebruikt worden. We schrijven geregeld over sprites, maar het onderwerp blijft interessant en we zijn dan ook blij, dat we hieronder een bijdrage kunnen plaatsen uit zeer deskundige bron, namelijk geschreven door Courbois, een bekende naam in CBM kringen.

Sprites of mobs



Sprites zijn zulke leuke en handige dingetjes, hoor je vaak. De echte Commodore 64 enthousiast noemt ze als eerste, wanneer je hem vraagt waarom hij een C-64 prefereert. Het nadeel bij Commodore is echter weer, dat niets eenvoudig gaat. Onze vrienden PEEK en POKE zijn helaas nodig om de sprites op het beeld en in beweging te krijgen.

Voordat een sprite gebruikt kan worden kom je om één ding niet heen: een sprite moet ontworpen worden. Net zoals een karakter. Een sprite is alleen iets groter. Om precies te zijn 24 bij 21 puntjes, dus bijna 3 bij 3 karakters. Totaal zijn dat 504 puntjes die op te slaan zijn in 63 bytes. Net zoals bij de karakters moeten we alles (bits-gewijs)optellen om de juiste bytes te krijgen om een sprite te definiëren.

regel 01
xxxxxxxx xxxxxxxx xxxxxxxx = XXX XXX XXX
regel 02
xxxxxxxx xxxxxxxx xxxxxxxx = XXX XXX XXX
regel 03
xxxxxxxx xxxxxxxx xxxxxxxx = XXX XXX XXX
etc.. tot ....
regel 21
xxxxxxxx xxxxxxxx xxxxxxxx = XXX XXX XXX

21 regels van 24 puntjes maken 21 maal 3 bytes. Bij het omrekenen is weer het meest linkse puntje 128 'waard' en het rechtse 1. De 24 bits maken zo 3 groepjes van 8 die elk een Byte zijn. 21 maal 3 is 63 Bytes per sprite.

Nu komt het POKE-kommando naar voren. De sprite moet ergens opgeslagen worden. Ergens in het geheugen komen 63 bytes te staan die later een sprite op het scherm zetten. Waar zetten we ze neer?

Geheugenplaats

Nu dat kan op een groot aantal plaatsen, 256 om precies te zijn. Of nog juister 4 maal 256 posities. Er is uit te rekenen waar dat is. De gehele geheugenruimte vanaf adres $0000 tot $4000 is daarvoor beschikbaar.

Verdelen we dit geheugen in blokjes van 64 bytes dan hebben we 256 mogelijkheden, dus theoretisch 256 sprites.

We geven die blokjes een nummer. Het eerste is 0 enz.... We kunnen zo eenvoudig de plaats vinden, waar de sprite komt. Willen we een sprite in blok 11 zetten dan is dat op geheugenplaats 11*64=704. Er vallen echter een paar plaatsen af. Zetten we bv. in het eerste blokje ($0000) iets, dan stopt de computer ermee. In die geheugenplaatsen zit vitale informatie zonder welke de computer niet kan werken.

Welke gebieden zijn dan wel beschikbaar?

Onderin het geheugen zijn er maar een 4-tal plaatsen. Het zijn de blokjes met de volgende geheugenlokaties:

Geheugen-lokaties	bloknummer
704-766	11
832-894	13 als cassette niet gebruikt wordt
896-958	14 als cassette niet gebruikt wordt
960-1022	15

Lager dan deze lokaties kan niet omdat dan het Basic kladblok vernietigd wordt. Hoger ook niet omdat op geheugenlokatie 1024 (spriteblok 16) de ruimte voor het Basic programma begint.

Is echter je programma niet te groot, bv. kleiner dan 10k, dan is er boven het programma nog wel plaats voor de spritedata. Net zoals karaktersets, die bv. op $3000 of $3800 kunnen beginnen is dit vaak ook de plaats waar de sprites staan. MAAR vergeet niet dat je het gebied moet beschermen tegen Basic overschrijving. Wil je bv. vanaf $3000 je sprites neerzetten dan moet je ingeven: **POKE 56,48: CLR.** Het spritebloknummer is dan 192.

Nu weten we dus waar we een sprite kunnen neerzetten en hoe die is opgebouwd. Maar daarmee zien we de sprite nog steeds niet op het scherm. We moeten de VIC-chip (de boosdoener) nog vertellen waar we ergens de sprite hebben neergezet. Wij weten dat wel, maar de VIC nog niet. Dat doen we nu weer met een poke. De VIC- chip kan de sprite vinden als op geheugenlokatie 2040 staat waar de sprite staat, het bloknummer.

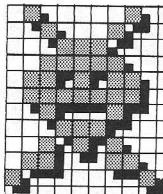
POKE 2040,192

Wil zeggen: sprite 0 staat op $3000. De 7 geheugenlokaties na 2040 (2041-2047) zijn voor de 7 andere sprites die mogelijk zijn. Willen we bv. dat de tweede sprite (sprite 1) het er hetzelfde uitziet als de eerste moeten we in geheugenlokatie 2041 ook 192 zetten. Willen we dat niet, dan moeten we het bloknummer van een andere sprite erin zetten.

Initialiseren van die 8 geheugenlokaties is niet nodig. Staan er vreemde waarden in dan geeft dat niet want...de sprite is nog steeds niet zichtbaar. De VIC-chip weet nu waar de sprite in het geheugen staat, echter nog niet waar die moet komen te staan op het scherm en...of de sprite wel zichtbaar moet zijn.

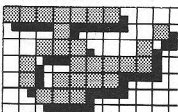
We moeten de sprite net zoals een lamp aanzetten. Daarvoor hebben we weer een

ingewikkelde regel nodig. In de VIC-chip zit 1 geheugenlokatie die de aan-uit-schakelaars voor alle 8 sprites bevat. Een sprite aanzetten moet dus met het OR-kommando, uitzetten met het AND-kommando:

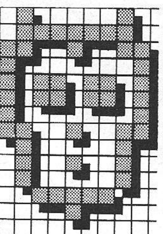


Aanzetten: POKE 53269, PEEK (53269) OR (2 " SPRITENUMMER)
waarbij SPRITENUMMER een getal tussen 0 en 7 kan zijn.

Uitzetten: POKE 53269, PEEK (53269) AND (255- 2 " SPRITENUMMER)

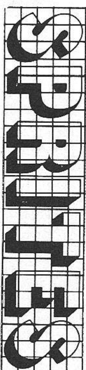


Willen we sprite 0, waarvan de plaats in geheugenlokatie 2040 staat, aanzetten dan doen we dat zo: **POKE 53269, PEEK (53269) OR (2 " 0).**



De sprite is nu zichtbaar. Als je de sprite gedefinieerd hebt dan zie je je sprite ergens staan. Zo niet dan zie je ergens een ongedefinieerde vlek staan. Dat is de toevallige inhoud van de geheugen lokaties aangegeven door geheugenlokatie 2040.

Om de sprite nu ergens neer te zetten op een bepaalde plaats moeten we de coordinaten doorgeven. Voor elke sprite zijn er twee geheugenplaatsen, een voor de x en een voor de y coordinaat.



De eerste sprite (sprite 0) heeft zijn x-coordinaat in 53249. De tweede sprite in de volgende twee .... enz.

Als laatste moeten we de sprite nog een kleur geven. Dit zijn dezelfde waarden als waarmee we het scherm van kleur veranderen, 0=zwart, 1=wit....enz. Voor elke sprite is er een kleur-register. Voor sprite 0 is dat geheugenlokatie 53287. De andere staan erna.

De x-coordinaat kan echter groter dan 255 worden. Het scherm is namelijk 320 breed en 200 hoog. Willen we nu een x-coordinaat groter dan 255 hebben moeten we die in twee bytes opslaan. De high-byte is altijd 1 of 0 (groter of kleiner dan 256). De low- byte kan alle waarden aannemen.

Willen we nu over de 256 heen dan moeten we 256 aftrekken van de coordinaat en de overgebleven waarde in het x-coordinaat register zetten. De high-byte (1) past in 1 bit, het is nooit groter dan 1. Dus ...waarom niet... alle 8 high-bits van alle 8 sprites kunnen in 1 byte. En het zou Commodore niet zijn...

De geheugenlokatie daarvoor is 53264 en er moet weer met OR en AND gewerkt worden.

Sprite X over x-coordinaat 255 :
POKE 53264, PEEK (53264) OR (2 " X)
Sprite X weer terug onder de 255 :
POKE 53264, PEEK (53264) AND (255 - 2 " X)

Nu kunnen een sprite op elke plaats neerzetten in elke kleur. Maar er zijn meer trucjes uit te halen met de sprite. Een lijstje daarvan :

- Een sprite kan voor of achter de karakters staan. Dit wordt bijgehouden in geheugenlokatie 53275. Weer 8 bits voor 8 sprites. Een 0 betekent voor de karakters, een 1 achter de karakters.

De POKE -regels worden dezelfde als net, maar nu met geheugenlokatie 53275.

- Een sprite kan zich 1 maal "uitrekken" in beide richtingen, de x-lengte en/of y-lengte wordt dan tweemaal zo groot. Ook dat gebeurt weer in slechts 2 geheugenplaatsen. Een 0 betekent niet "ge-expandeerd" en een 1 wil zeggen "ge-expandeerd" oftewel groter.

De x-coordinaat staat in geheugenlokatie 53277 en de y-coordinaat in geheugenplaats 53271. Weer dezelfde poke-regels als hiervoor.

Met deze geheugenlokaties kunnen we nu een sprite over het scherm laten lopen, een kleur geven en de sprite groter en kleiner maken. We kunnen ook een sprite achter karakters verbergen en voor de karakters laten bewegen. Je kan echter nog meer met sprites. B.v. registreren dat twee sprites elkaar raken, dat sprites karakters raken, multi color sprites enz, enz.

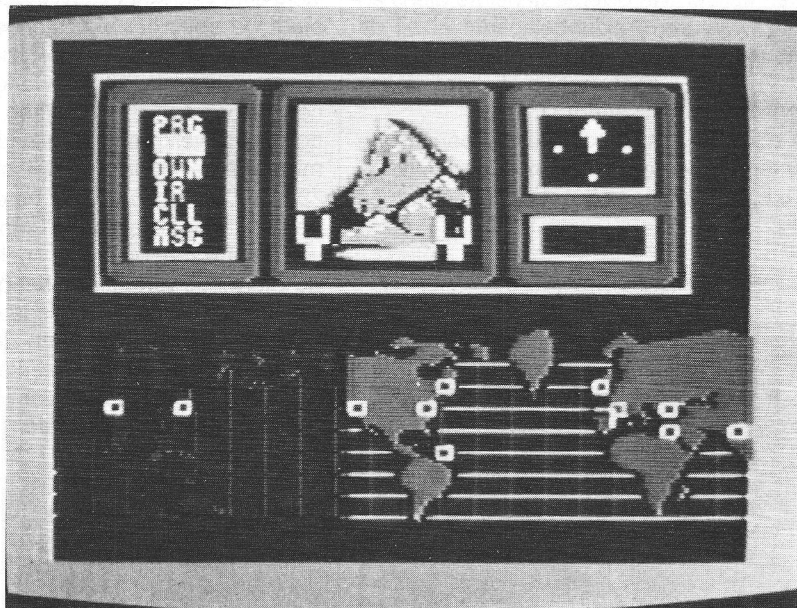
We gaan daar niet uitgebreid op in, dat is iets voor een sprite-cursus. Wel volgt hier onder een overzicht van alle geheugenplaatsen die wat met sprites te maken hebben.

53248 t/m 53263	x en y coordinaten van de 8 sprites
53264	high-bit van de x-coordinaat
53269	aan en uitzetten van de 8 sprites
53271	y-expand
53275	voor of achter de karakters
53276	multicolor aan of uit
53277	x-expand
53278	botsing tussen 2 sprites
53279	botsing tussen sprite en karakters
53285	multicolor kleur 2
53286	multicolor kleur 3
53287 t/m 53294	kleuren voor de 8 sprites

Een overzichtje :

	0	1	2	3	4	5	6	7
Bloknummer	2040	2041	2042	2043	2044	2045	2046	2047
Kleur	53287	53288	53289	53390	53291	53292	53293	53294
X-coordinaat	53248	53250	53252	53254	53256	53258	53260	53262
Y-coordinaat	53249	53251	53253	53255	53257	53259	53261	53263

Er kan heel wat meer met je 64 dan alleen maar braaf spelen, ook qua criminele activiteiten kun je er mee uit de voeten. Van misdaadspelletjes tot aan het allernieuwste Hacker, een spel zonder spelregels, dat als doel heeft om in andere computers in te breken.



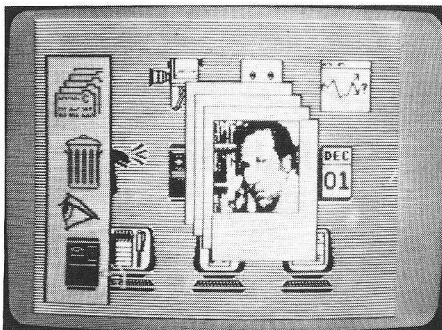
MISDAAD PER 64

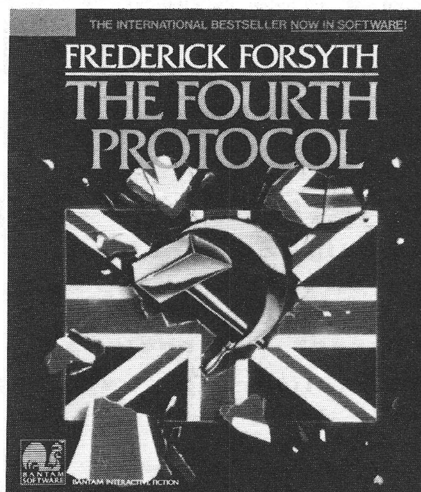
Ulco D. Schuurmans

Uit de VS nadert een ware misdaadgolf op het gebied van computerspelletjes en adventures. Betrof het voorheen slechts het ordinaire schietwerk, nu stellen de softwarehuizen aanzienlijk hogere eisen aan hun produkten en sommige zijn zelfs educatief te noemen. Ulco Schuurmans neemt u mee naar de wereld van sluwe schurken,te ontrafelen raadsels en dreigend tikkende bommen.

Detectivespelen

We kennen ze allemaal wel. De spelletjes met integere harde speurders en de snode schurken die de wet met voeten treden. In spelletjes als **"Gumshoe"(SOFTAID)** staan ze elkaar danig naar het leven. Een rap vingertje op de vuurknop en snelle rukjes aan de spelpook brengen de speler een heel eind. Snelheid is daarbij veel belangrijker als inzicht. Games als **"A View To A Kill"** brachten daar al verandering in door behalve het actie- ook de nodige dosis speurwerk ın te bouwen. Een nieuwe ontwikkeling is de adventureroman. Gerenommeerde detective- en spionageschrijvers vertrouwen nu hun verhalen aan de computer toe i.p.v. het papier. "The Fourth Protocol" van Frederick Forsyth is daarvan een goed voorbeeld.





The Fourth Protocol

Het vierde Protocol bevat alle elementen van de avonturen & misdaadroman. Een dreigende coup, een op scherp staande en aftellende atoombom, verraad, politieke intriges en geheime agenten. De speler dient in de rol van John Preston het uitttreden van Engeland uit de NATO te voorkomen. Vreemde inlichtingendiensten smeden een komplot om via een "ongelukje" met een nucleaire bom de betrekkingen tussen GB en de VS te verzieken. Het spel bestaat uit drie onderdelen. In deel een dient men de verrader te ontmaskeren. John Preston onderneemt hierbij alle actie van uit zijn MI5 buro. Dossiers naslaan, telefoneren, personen laten schaduwen. De handelingen blijven echter wat statisch, het is echt meer een tekstavontuur met grafische support.

In deel twee komt er meer leven in de brouwerij. De bom moet gevonden worden voordat deze af gaat. De actie is hier jammer genoeg symboolgestuurd, maar boeit wel. Na dat de speler een symbool gekozen heeft openen zich meerdere beeldschermvensters met submenu's. Ook wordt de reactie op de ondernomen actie in de beeldvensters weergegeven. John Preston kan bij zijn naspeuringen gebruik maken van de taxi, de ondergrondse of de benenwagen. Alle graphics zijn fraai gedetailleerd.

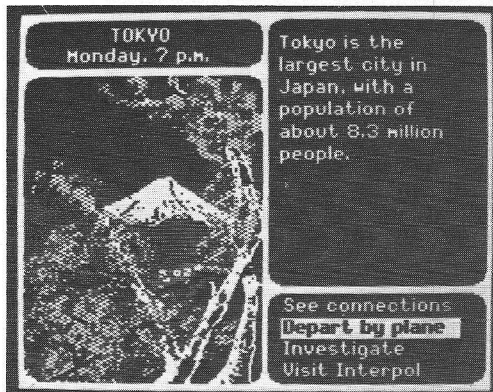
Het derde deel,waarin de gevonden bom ontmanteld dient te worden,is wat teleurstellend. De graphics zijn matig, de adventureteksten niet meer dan doorsnee en men moet zo nodig ook nog wat KGB-agenten afschieten. Over het geheel genomen toch veel meer dan het gewone schietspelletje of adventure.

Voor de C-64 als cassette van Hutchkinson Software.

Schurkenjacht

Ging het bij Het Vierde Protokol meer om het opsporen van de bom, in **"Where in the World is Carmen Sandiego"** is het te doen om de jacht op een supergangster. Maar wat een achtervolging. Over een grootdeel van de wereld, met veel educatieve informatie en fraaie graphics. Bovendien moet men in dit zeer geraffineerde detectivespel

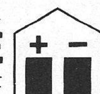
echt logisch denken en kombineren. Carmen Sandiego is een vrouwelijke schurk van kaliber. Na diverse geheime diensten als twee- en driedubbelagente al bij de neus gehad te hebben opereert zij nu zelfstandig, kompleet met haar eigen gangsterorganisatie V.I.L.E.
De speler komt in actie na een bericht van Interpol over de laatste misdaad van V.I-.L.E. Het begin van het spoor naar Carmen moet uit de getuigenverklaringen komen en dat is niet zo gemakkelijk. Dan begint de jacht over vele steden en landen. Elke stad biedt de speurder vier menuopties: Met **"Investigate"** verschaft men zich de nodige gegevens via getuigenverklaringen en het bezoek aan drie plaatsen. Komt u er achter na welk gebied de daders van de misdaad gevlucht zijn dan staan onder **"Show Connections"** alle bereikbare steden. Via **"Depart by Plane"** kan de detective dan afreizen. Meent de speler voldoende de informatie vergaard te hebben dan kan via **"Visit Interpol"** een misdaadcomputer geraadpleegd worden. Aan de hand van zaken als bijzondere kenmerken, hobbies, haarkleur e.d. krijgt men een lijst van mogelijk passende booswichten.



Het leuke van deze wereldjacht is dat men spelenderwijs veel geografische kennis opdoet. De speler wordt naast de schermteksten gesteund door de bijgeleverde Wereld Almanac (Amerikaanse Versie). Een grote variëteit aan daders en misdaden,en de mogelijkheid om promotie te maken houden het spel lang boeiend. Op disk voor C-64.

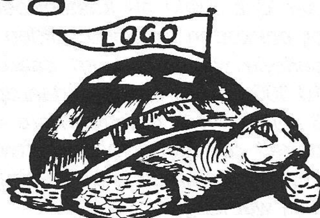
Gamegolf

De hier genoemde detectivewerkjes vormen slechts het begin van een nieuwe gamegolf. **Perry Mason** is al aangekondigd en de ware liefhebber kan alvast terecht bij adventures als "Amazon", "Dragonworld", "Murder on the Zinderneuf",en de Infocom-spelen "The Wittness","Suspect" en "Deadline".
Van een misdadig karakter met een nog duidelijker computertintje kunnen we spreken bij **Hacker**, het nieuwe spel van Activision. Dat wordt zonder handleiding, zonder enige aanwijzing geleverd, we willen alleen vertellen, dat het gaat om het inbreken in allerlei andere computers via codewoorden en Log-On procedures en dat het als adventure uiterst boeiend en quasi- realistisch is. Maar liefst niet echt met een modem op deze manier aan de gang.
We hebben Hacker geprobeerd en eerlijk gezegd zit het zo vol met computer"slang", dat je erg goed in je enegels moet zitten om je een weg te banen door het doolhof van de vreemde computers. Bijna 70% van de schermen vragen om reakties op tekst. ●



Van de Logo-Stichting Nederland kregen we wat commentaar op onze reeks artikelen over Logo. Men vindt, dat we die taal toch nog niet serieus genoeg zien. We bieden ze graag wat ruimte voor hun visie.

Doorbraak van Logo



Succes voor Logo nu een feit

Logo lijkt een beetje het slachtoffer van haar eigen succes (voornamelijk op de Amerikaanse basisschool) te worden. Het heeft daardoor een reputatie verworven als een triviale taal voor jonge kinderen, waarmee ze leuk op de TV kunnen tekenen en knutselen.

LOGO is veruit de krachtigste programmeertaal, die op dit moment voor huiscomputers ter beschikking is.

Wat maakt een programmeertaal krachtig? Niet dat je er unieke dingen mee kunt doen, die niet met een andere taal mogelijk zijn. Wat dat betreft ontlopen programmeertalen elkaar niet zo veel: wat je in LOGO kunt programmeren, gaat ook wel met Pascal of Basic en andersom. De kracht van een taal moet worden afgemeten aan de scheiding tussen taal en probleem. In hoeverre helpt de taal om, je te concentreren op het actuele probleem dat je wilt oplossen, eerder dan dat je je zorgen moet maken over de dwangmatigheden van de taal.

Compositie van funkties

Het is niet toevallig dat LOGO krachtiger is dan BASIC of Pascal. Fortran werd bedacht voordat de mathematisch basis van het computerprogrammeren goed werd begrepen. Daardoor reflecteert Fortran de mogelijkheden (en ontoereikendheden) van de destijds beschikbare computers. Krachtiger talen zijn gebaseerd op een of ander mathematisch model van rekenen en gebruiken dat model op een consistente manier. Zo is LOGO gebaseerd op het idee van een compositie van funkties. LOGO is eigenlijk een dialect van LISP, de taal die ook nu gebruikt wordt in de meest geavanceerde onderzoeksprojecten in computerwetenschappen, met name bij kunstmatige intelligentie.

Programmeren is in wezen mathematisch, of formeel denken. Leren programmeren in LOGO, of je dat nu doet met de (teken-) sachildpad, zijn klonen (de sproken), of met de professionele programmeereigenschappen, het is op z'n minst een vorm van intellectueel leren en een goede voorbereiding op functionele talen en kunstmatige intelligentie.

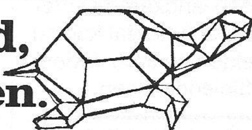
Tot op heden behandelen vrijwel alle LOGO-boeken LOGO's tekenmogelijkheden: schildpadplaatjes. Deze aardige plaatjes vormen slechts een deel van het totale LOGO-plaatje. Naast de **Schildpadwereld** zijn er de **Woordwereld**, de **Muziekwereld** en de **Animatie en Simulatiewereld**. De elektronische knutseldoos van Logo biedt dus veel meer, het leren houdt niet op bij mooie grafische kunstjes, alle normale computertoepassingen zijn ook in Logo opgenomen.

Computer Science LOGO Style van Brian Harvey poogt LOGO de plaats te geven die het verdient. Harvey zegt: "Iemand die nooit geprogrammeerd heeft in LOGO, mist de achtergrond om de argumenten voor het gebruik van die taal echt te kunnen begrijpen." Harvey studeerde o.a. aan het MIT, en was een van de ontwerpers van de Apple en Atari-LOGO's. Dat zijn LCSI-versies van LOGO, dat ook nog andere versies kent (Dr. LOGO, ExperLogo, TI-Logo enz).

LCSI uit Canada heeft nu ook een LOGO voor de MSX-computers gecompileerd, die door Philips in Nederlandse vertaling wordt uitgebracht. En dat nu zou wel eens de doorbraak kunnen worden voor LOGO in Nederland. Of het onderwijs met LOGO gediend is? LOGO-adepten weten het zeker, anderen moeten nog overtuigd worden. Echt onderzoek naar de waarde van LOGO is in Nederland echter nog niet gedaan. Hopelijk komt daar snel verandering in.

Harvey, B. Computer Science LOGO Style. The MIT Press. 1985. ISBN 0-262580721. ●

Logo in Nederland, de stand van zaken.



Al sinds 1982 propageert de LOGO-stichting Nederland het gebruik van LOGO in het onderwijs. Om dit te bevorderen, ontwikkelt men in Nijmegen de LCN LOGO, een Nederlandstalige LOGO geschikt voor de meest verkochte computers. Voorjaar 1986 zal dit geavanceerde softwarepakket leverbaar zijn.

De stichting LOGO K in Zwolle organiseert al jaar en dag LOGO zomerkampen. Daarnaast zijn diverse LOGO-centra regionaal werkzaam. LOGO maakt ook - zeer bescheiden - deel uit van de nascholingscursussen voor leerkrachten aan het basisonderwijs. Het PBNA heeft een LOGO-programmeercursus in voorbereiding.

Voor vrijwel elke type computer is een LOGO-pakket leverbaar (meestal in het Engels). In het Nederlands kan LOGO op de Commodore 64 (op disk, uitgave Malmberg Den Bosch) en Philips heeft voor de MSX een LOGO-module op de markt gebracht. Omdat de handleiding geen aandacht schenkt aan **woord- en lijstbewerking** is hiervoor door het LOGO Centrum Ede (Annadaal 96, 6715 JC Ede 08380-21306) een speciaal **Doe Het Zelf-praktikum** ontwikkeld. Voor het onderwijs zijn speciale oefenboekjes ter beschikking. Ook zijn er al taal- en rekenprogramma's in LOGO geschreven en dus eenvoudig aan eigen wensen aan te passen.

De **AULA pocket 740** van Pinxteren en Ringelberg geeft een goede inleiding in LOGO.

Het boek **"Computer en Kinderen"**, van Seymour Papert (uitg. B.Bakker), de bedenker van LOGO levert de theoretische uitgangspunten.

Binnenkort komen nog enige nieuwe boeken over LOGO op de markt: oa. **LOGOlogie**, computerwetenschap in LOGO-stijl, mikt vooral op de middelbare scholier, die genoeg heeft van z'n Basicproblemen. Het landelijk informatiepunt kan eventueel verder helpen: u kunt bellen naar 08380-21306.



LUC SALA'S DATAKOLOM

Het belastingbiljet in de bekende blauwe envelop betekent voor veel mensen weer het begin van een periode puzzelen, zoeken, en gokken. Wat zal de belastingdienst nog wel en wat zal men zeker niet door de vingers zien aan buitengewone lasten, verwervingskosten en reiskosten. Weer een heel gezoek in de schoenendozen, klappers, mappen en tussen de papierkraam van verzekeringen en bank- en girobescheiden. Gelukkig kunnen we bij dat invullen de computer gebruiken, daarmee is het rekenwerk wat eenvoudiger geworden en komt de zaak ook overzichtelijk op een uitdraai te staan.

Belastingperikelen

Ik wil hierbij niet ingaan op de merites van de diverse belastingprogramma's, maar mijn mening geven over de afschrijving op computerapparatuur en software. Dat is natuurlijk voornamelijk voor het bedrijfsleven van belang, maar ook de particulier heeft er wel mee te maken. Afschrijving van apparatuur, die is aangeschaft voor kantoor aan huis, voor studie of opleiding of om neveninkomsten te verwerven, is van belang voor de belastingaangifte of bij het opmaken van de balans en andere jaarstukken.

De meeste belastingadviseurs en belastinggidsen adviseren een periode van 5 jaar als afschrijftermijn voor computerapparatuur en software. Nu even afgezien van de vraag, of daarbij wel of niet van lineaire afschrijving (gelijke delen per jaar) en van historische kostprijs danwel vervangingswaarde moet worden uitgegaan, is dat naar mijn mening een te lange termijn.

Afschrijving op hardware en software is in het algemeen te conservatief.

Gezien de trends van de afgelopen 5 jaar, waarbij enerzijds de aanschaffingsprijs van zowel hardware als software in een vrij snel tempo is gedaald, als ook de economische levensduur van die zaken erg kort aan het worden is, is een vrij korte termijn verdedigbaar.

Het is mogelijk een algemene indicatie te geven, hoewel in individuele gevallen afwijkingen mogelijk zijn.

Bij hardware tot ongeveer 3.000 gulden per systeem is een afschrijving van 2 jaar aan te nemen, bij hardware tot 15.000 gulden per systeem 3 jaar, terwijl daarboven slechts in uitzonderingsgevallen een afschrijvingstermijn van meer dan 4 jaar dient te worden gehanteerd.

Voor software bij een pakketprijs tot 500 gulden is de periode van afschrijving 1 jaar (dus direkt als kostenpost nemen), bij pakketten tot 2500 gulden is er een periode van 2 jaar en daarboven tot 10.000 gulden voor standaardpakketten 3 jaar en voor maatsoftware 4 jaar. Dit betreft overigens micro-software, voor grotere systemen is de afschrijvingstermijn wat langer.

Als onderbouwing van de bovengenoemde termijnen eerst wat voorbeelden.

De **aanschafprijs** *is drastisch aan het dalen. Wie een jaar geleden een MSX computer aanschafte voor rond de 1200 gulden en nu ziet, dat die prijs tenminste gehalveerd is, begrijpt dat afschrijving en waardering op basis van de vervangingswaarde een periode van 2 jaar meer dan aannemelijk maakt.*

Ook heeft hij misschien gemerkt, dat de **technische levensduur** *van apparaten aanzienlijk korter is geworden. Nier zozeer, omdat ze kwalitatief slecht zijn, maar omdat reparatie de moeite niet meer loont en al bij een bescheiden mankement vervanging verstandiger en goedkoper is en de feitelijke levensduur dus bepaald wordt door de MTBF (gemiddelde tijd tussen storingen).*

De **economische levensduur** *is ook maar zeer beperkt, vervanging is al snel goedkoper dan doorgaan met het oude spul. Dat geldt zeker als we de* **ergonomische** levensduur daarin betrekken. Enige jaren geleden kon er meer of juist minder dan nu verantwoord is. Bijvoorbeeld tekstverwerking met een 40 koloms beeldscherm, een langzame printer en beperkte opslagcapaciteit, dat kan nu toch niet meer. Er gelden nu toch andere ergonomische en efficiency eisen.

De ergonomische levensduur is beperkt, is het nog wel wenselijk met bv. een 40 koloms scherm te werken.

Dus dat oude systeem kan misschien nog als gift naar een basisschool in de buurt, maar om het tegen 80 of 60% van de aanschaffingsprijs in de boeken te houden, is wat onzorgvuldig.

De eisen aan apparatuur en toepassingen nemen zo toe, dat waar 128 KB een jaar geleden voor een PC heel leuk was, men nu met minder dan 640 KB een heleboel programma's niet eens goed kan draaien en we volgend jaar echt in MB's moeten rekenen voor gewone PC's. Hetzelfde geldt voor de interfaces, de beeldschermkwaliteit en het geluidsniveau en afdrukkwaliteit van de printers. Het is niet meer verantwoord om duur personeel te laten wachten op een 12 t/s printertje met een 70 dB geluidsniveau. Wanneer een nieuwere machine de wachttijden, het ongemak, de risico's van verloren files of onvoldoende backup's vermindert, en bovendien niet duurder is dan zijn voorganger, dan is het pleit snel beslecht. De oude computer of printer promoveert naar een niet-gebruikt bureau of tot kastcomputer. De inruilwaarde is zeer gering, al kan men net als bij auto's optisch nog wel eens een aardige inruilprijs denken te krijgen. Meestal laat men het ding, dat toch pas een of twee jaar oud is, maar staan, maar het gebruik ervan is minimaal. Wie wil weten, wat tweedehands huiscomputers waard zijn, hoeft maar te kijken naar de advertenties in de computerrubrieken en bladen, na een jaar is 30% al heel aardig, terwijl men voor de extra's, accessoires en software al helemaal niets overheeft.

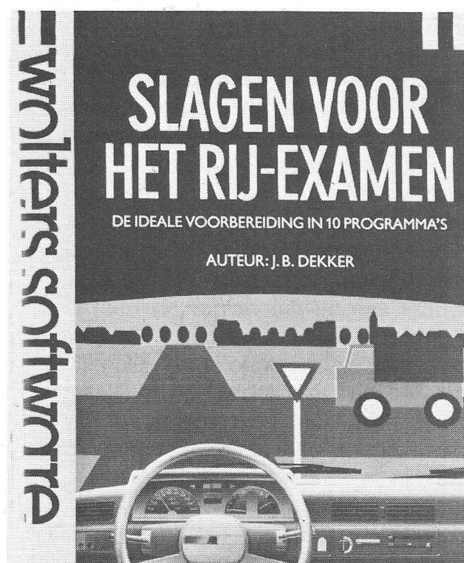
En dat is begrijpelijk, want de prijzen dalen ook nog eens in recordtempo. Wie vorig jaar een normale 256 KB floppy disk MS-DOS PC wilde hebben, was daar minimaal 8600 gulden voor kwijt. Nu kan men een middenklasse systeem voor 3900 gulden en een goedkoop merk voor minder dan 3000 gulden aanschaffen, terwijl die prijzen volgend jaar echt nog 30% lager liggen. Nu al valt te verwachten, dat een 80286 AT micro voor het eind van het jaar voor minder dan 9.000 gulden, inclusief 10 MB hard disk, te koop is, terwijl men nu nog minstens 16.000 gulden kwijt is.

Ook bij de software geldt iets dergelijks. Ook daar volgen nieuwe versies en prijsdalingen elkaar met grote regelmaat op en wie nu nog 2000 gulden betaalt voor bv. Wordstar is een wel erg gemakkelijke klant. En mede door de activiteiten van bv. Borland in de US zal de prijsdaling zich nog wel voortzetten. Bovendien maken nieuwe verkooptechnieken, zoals site-licensing voor grote bedrijven, de waarde van de aparte software-pakketten geringer. En bij software is de economische veroudering nog veel sneller dan bij hardware, de toenemende efficiency bij specifieke toepassingen maakt aanschaf van nieuwere versies en nieuwere pakketten al snel verantwoord.

Alles bij elkaar vind ik, dat er zeer goede argumenten zijn, om de afschrijving van hard- en software behoorlijk te versnellen. Dat kan (op korte termijn) wat financiële ruimte scheppen, die dan misschien weer gebruikt kan worden om de efficiency te verhogen met betere spullen.

Ir. L.Sala

Je voorbereiden op de rijexamen theorie of de verkeerskennis eens opfrissen met de Commodore? Dat kan met twee nieuwe educatieve programma's van Wolters Software: "Slagen voor het Rij-Examen" en het meer speelse "Verkeers Rally".In deze korte softwarebespreking maakt u kennis met een nieuwe aanpak van het verkeersonderwijs.



Rij-examen per computer?

Verkeerssimulatie

De grafische mogelijkheden van de C-64/C128 (in de 64 mode) lenen zich uitstekend voor het simuleren van verkeerssituaties. Kruisingen, de gedragingen van andere verkeersdeelnemers, verkeersborden en - lichten, reactietesten, verschillende geluiden, zelf rijden, diverse voertuigen, noem maar op. Uw computer tekent ze snel op het monitor- of tv-scherm.

Dergelijke gedetailleerde grafische voorstellingen kunnen natuurlijk ook gemakkelijk als onderwerp voor een lessituatie gebruikt worden. U krijgt bijvoorbeeld een kruising op het scherm en de computer stelt de vraag:
"Welke van de naderende verkeersdeelnemers heeft voorrang?", gevolgd door een aantal mogelijkheden of ja/nee- opties.De gebruiker kan dan zijn/haar verkeerskennis toetsen door het juiste (?) antwoord in te typen. De C-64/128 controleert aan de hand van de programmagegevens of dit inderdaad het juiste antwoord is. Zo ja, dan de volgende vraag. Zo nee, dan volgt een foutmelding met eventuele nadere uitleg. Kortom onderwijs in optima forma.
Slagen voor het Rijexamen en **Verkeersrally** bieden het voordeel dat de rijexaminant, scholier of verkeersdeelnemer die zijn kennis weer eens wat wil opfrissen, niet voortdurend terug hoeft naar die saaie voorbeeldboekjes. Je voert, gezeten in je eigen luie stoel, een soort leraar-leerling dialoog met de computer. Het bijkomstige spelelement maakt bovendien het leerproces aanzienlijk gemakkelijker. Hoewel beide programma's het echte theorieonderricht natuurlijk niet kunnen vervangen geven zij toch een aardige ruggesteun waar ook geroutineerde weggebruikers nog veel plezier aan kunnen beleven. Voor echt professionele rijschoolprogramma's zult u een dure beeldplaat met laserafspeler moeten aanschaffen. Daarmee zijn "real life" verkeerssituaties te simuleren. Wellicht iets voor rijscholen, maar zeker (nog) niet voor de thuiscomputeraar.

Slagen voor het Rij-Examen

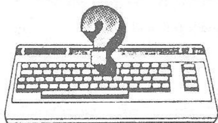
Slagen voor het Rij-Examen is een primair educatief programma voor de Commodore 64. Het bestaat uit twee delen van elk vijf programma's. In deel 1 wordt het rijgedrag op kruisingen behandeld. Daarbij leert u wat in de gegeven gevallen het meest correcte en veilige rijgedrag is. Een leuke bijkomstigheid is dat men gelijktijdig de reactiesnelheid en het waarnemingsvermogen kan testen. De tweede groep van vijf programma's legt de gebruiker meer dan 120 verschillende verkeerssituaties voor. De gegeven antwoorden worden waar nodig toegelicht en gemaakte fouten later opnieuw op het scherm aangeboden. Alle verkeerssituaties richten zich op het theorie-examen voor het rijbewijs. Zij zijn dan ook gebaseerd op het boek "Slagen in het Verkeer", een uitgave van de ANWB, het Centraal Bureau Rijvaardigheidsbewijzen en Veilig Verkeer Nederland.
Wij gaan even kort in op de verschillende programmaonderdelen. Het programma "Gedrag op Kruisingen" bestaat uit de volgende vijf delen:
❶ Algemene aanwijzingen met het gedrag op kruisingen en begeleidende vragen, een reactietest en een ogentest.
❷ Voorrang: vragen en examenvragen, en borden.
❸ Verkeerslichten; Praktijk verkeerslichten. Bijzondere weggebruikers. Bestuurders. Nog wat over voorrang. Hinderen.
❹ Richting veranderen;Praktijk, hinderen en examenvragen.
❺ Waarnemingstest en kleurentest.
Bij de reactietest dient u tijdig een functietoets in te drukken als een groene auto op het zwarte beeldscherm verschijnt. Daarna komt uw reactietijd met beoordeling op de monitor of tv. Voor de ogentest meet u eerst de breedte van het blauwe scherm.Na het intypen van dit gegeven toont de Commodore de gewenste testafstand tot het scherm. De tester, liefst een ander dan de proefpersoon, laat d.m.v. een functietoets gedurende vijf seconden een auto met kentekenplaat zien.Vervolgens biedt de computer zeven mogelijkheden waaruit de proefpersoon het juiste kenteken moet kiezen.
Tijdens het eigenlijke "lessen" komt u situatievragen en rijopdrachten tegen. Bij de situatievragen moet men uit een meerkeuzemenu kiezen of ja/nee beantwoorden. De situatievragen zijn behoorlijk realistisch en worden soms zelfs met geluid begeleid. Wist u dat politie en brandweer een tweetonige- en de ambulance een drietonige hoorn voert? Zo nee, of kon u het verschil nooit duidelijk horen, dan valt dit middels de juiste hoornsignalen bij het desbetreffende voertuig te beluisteren. Politie en brandweer klinken daarbij heel natuurge-

trouw. De ambulancesirene is weliswaar drietonig maar niet echt HiFi.
De rijopdrachten zijn zowel met joystick(poort 2) als toetsenbord uitvoerbaar. De bewegingen verlopen zoals wij van de doorsnee videospellen gewend zijn enigzins hoekig en houterig. Zij zijn echter voldoende illustratief voor de gegeven gedragsregels en verkeerssituaties.
In het theorie-examen (andere cassette of andere zijde van de diskette) bent u de bestuurder van een witte personenauto met het opschrift "IK". De vragen moeten met J(=ja) en N(=nee) beantwoord worden. Het geheel lijkt op de situatie bij de diapresentatie op het theorie-examen. Natuurlijk zijn de diabeelden van het CBR-examen veel gedetailleerder en krijgt u ook de uitslag niet direkt te horen. Slagen voor het Rij-Examen is echter een privélessituatie en dat maakt weer veel goed. Bovendien krijgt de leerling de fout beantwoorde situatie later weer terug als alle overige vragen zijn afgewerkt. Aan het eind van elk programmadeel volgt een foutenscore.
Wolters Software is er binnen de beperkingen van de huiscomputer toch aardig in geslaagd om een leuke introductie tot het theoretisch rijexamen te maken. De inhoud is niet voldoende maar vormt een goede aanzet en is tevens geschikt als geheugenopfrisser voor de reeds geslaagde weggebruikers. De graphics kunnen beter, want zij geven soms aanleiding tot onnodige misverstanden. Misschien iets voor een volgende versie?

Verkeers Rally

Het programma Verkeers Rally is veel speelser van opzet dan Slagen voor het Rij-Examen. De opdracht is binnen een uur zes zenders (een zwart rechthoekje op het trottoir) binnen een onbekende wereldstad op te sporen. Daarbij wordt de speler natuurlijk met een aantal moeilijkheden geconfronteerd: iedere op te sporen zender (NB er is er altijd maar één in de lucht!) heeft voor slechts tien minuten batterijspanning. Lukt het niet om de zender tijdig op te sporen, dan zal dat ook niet meer lukken. Tenzij u opnieuw begint. Een tweede moeilijkheid vormt het vinden van de weg. Plattegronden, signaalborden, puzzelgebouwen en sonarstations geven daartoe aanwijzingen die de speler nauwlettend in de gaten dient te houden. Een druk op de vuurknop na het parkeren bij een wijk- plattegrond, sonarstation of puzzelgebouw activeert de desbetreffende schermaanwijzingen.
Dan is er ook nog het benzineprobleem. Tijdig tanken brengt vanzelfsprekend de oplossing.
Waar blijft nu het educatieve verkeerselement zult u zich inmiddels afvragen? Het verkeersonderricht schuilt in de tussen de verschillende spelsituaties door gestelde voorrangsvragen. M.b.v. de joystick wijst u de weggebruiker met voorrang aan. Foute antwoorden verhinderen de voortgang en kosten tijd. Verkeersovertredingen en botsingen leveren een flink aantal strafseconden op. Als laatste doet het kaartlezen en positie inschatten een stevig beroep op uw orientatievermogen.
Een leuk spel met educatieve kanten om af en toe de verkeerskennis eens wat te beproeven. Jammer dat de bewegingen soms weer wat te houterig aan doen. De goede verzorging van de graphics verzacht gelukkig de minder vloeiende bewegingen. Ook hier mogen wij bij toekomstige versies misschien meer verwachten.
Slagen voor het Rij-examen is op cassette (2 stuks) en diskette (tweezijdige) leverbaar en kost ƒ 59,-. Verkeers Rally kost op cassette en diskette ƒ 49,-. ●

VRAGEN VAN GEBRUIKERS



Microfoon op de C-64

Wilco van Sonsbeek uit Rosmalen zit er al een poos te dubben over de mogelijkheid om een microfoon op de C-64 aan te sluiten. Hij vraagt of dit eigenlijk wel kan.

De aansluitingen op de C-64, die betrekking hebben op Audio en Video zijn inderdaad voorzien van een aparte ingang voor Audio In. Deze zit op pin 5 (en aarde op pin 2) van de DIN-aansluiting. Door een voorversterkt geluidssignaal hierdoor naar binnen te sturen vanuit bv. de Hi-Fi installatie kun je dit signaal mee laten gaan met de door de SID-chip zelf gemaakte geluiden. Ook is het (beperkt) mogelijk om externe syntheziser- en andere geluiden via dit kanaal toe te voegen aan de door de C-64 opgewekte geluidenstroom.

Printer-interfacing

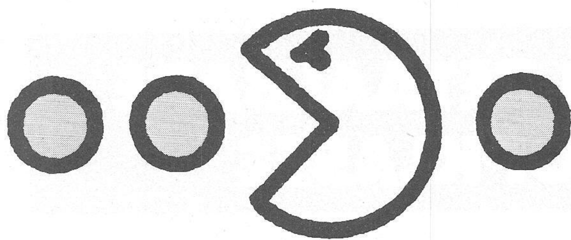
De vragen over het aansluiten van allerlei niet-Commodore printers aan de C-64 en C-16 zijn niet van de lucht. Het blijkt dat ook de handel hier eigenlijk geen raad mee weet. Men wil graag hardware verkopen, maar daarna zoekt de gebruiker het zelf maar uit. Wat is nu eigenlijk aan te sluiten op wat?

Op de Commodore C-serie (C16/C64/C128) zijn in feite allerlei goede printers aan te sluiten, als er bij de printers maar gebruik wordt gemaakt van de standaard data-overdracht. We kennen daarbij de RS232-C seriële en de Parallel-Centronics gestuurde printers.
Geen van deze zijn echter zonder meer op de Commodore aan te sluiten. En het is ook niet alleen een kwestie van een andere kabel en/of plug. Om met de printers uit deze categorieën te kunnen werken heb je een **interface** nodig, die beide apparaten op de juiste manier verbindt.
Een interface moet je zien als een soort signalen-tolk, die de printer-signalen uit de C-64, die eigenlijk bestemd zijn voor een Commodore printer, vertaalt naar de signalen die een RS232 of Centronics printer nodig heeft. Er zijn talloze van deze apparaatjes op de markt, die in prijs varieren van zo'n 100 tot 500 gulden. De prijs heeft voornamelijk te maken met het aantal mogelijkheden dat in zo'n print is ingebouwd. Bij de duurdere typen heb je vaak ook de optie om de grafische Commodore-karakters te kunnen printen. Ook bepaalde (betere) toolkits zijn vaak uitgerust met één of andere interface mogelijkheid.
Het is dus zaak om niet alleen bij de aanschaf van 'vreemde' printers, maar ook bij het kopen van de toolkits te letten op de diverse mogelijkheden. In het algemeen geldt dat zowel Centronics als RS232-C printers zonder al te veel moeite en kosten samen met de Commodore C-serie kunnen worden gebruikt, je moet er daarbij wel van uit gaan, dat deze printers niet voor 100% te vergelijken zijn met de printers van eigen merk. De kwaliteit laten we liever buiten beschouwing.

Kraakvrijheid

Één van de lezers stuurt een verzoek in om de mogelijkheid te onderzoeken het programma **Colossus Chess 2.0** *via een turbo-save/load cyclus op een snellere manier te kunnen gebruiken. Hij is er zelf al een paar maanden mee bezig, maar kan er niet uitkomen.*

Beste meneer Jansen, als we op een vraag als deze een toepasbaar antwoord geven, maken we ons schuldig aan piraterij. Want als zo'n programma éénmaal is gecopiëerd dan is er geen enkele garantie meer, dat er niet meerdere exemplaren van in omloop komen, zonder dat de maker/verkoper zijn rechtmatige beloning hiervoor ontvangt. Kortom, het is lastig, dat het programma er zo lang over doet om te laden, maar aan de andere kant zijn de auteurs zo verstandig geweest, naast het normale RUN-gedeelte in de RAM van de C-64 ook nog een min of meer geheim stuk data ergens in het geheugen weg te zetten. Bij het saven wordt dit 'onvindbare' gedeelte niet meegenomen. Tijdens het runnen van de illegale kopie kijkt de software na elke zet of het stukje geheime data wel aanwezig is. Zo niet, dan volgt er een hang-up. Ik kan het de auteurs niet kwalijk nemen. ●



Basic Min

Een rubriek van Nico Baaijens me

Help! De Basic Miniaturist van Commodore Info verzuipt! In het aantal inzendingen voor deze ongelofelijk populaire rubriek gelukkig en dat betekent weer smikkelen en smullen voor een ieder die meer uit zijn computer wil halen. Laat ik niet te lang inleiden en snel over gaan tot de orde van de maand.

Lettermonitor

Lars van der Steen opent de revue met iets leuks. Een lettermonitor, die na het invoeren van een getal het bijbehorende letterteken grafisch weergeeft in een matrix. Gewoon intypen en proberen.

```
10 INPUTP:V=56334:POKEV,0:POKE1,179:FORQ=0TO7:
   PRINT:FORI=0TO7
20 PRINTCHR$(46+SGN(PEEK(P*8+Q+53248)AND2<PIJL
   OMHOOG>(7-I))*67);:NEXTI,Q:POKE1,183:POKEV,1
```

Handig

Een handige routine stuurde Job ten Hagen uit Huissen op. Na RUN gebeurt er niets, maar nu is de zo zelden gebruikte F1 toets van kracht. Hij verandert de randkleur. Als in regel 30 de waarde 32 in 33 wordt veranderd, zet F1 telkens een andere schermkleur op.

```
20 FORT=0TO24:READA:POKE700+T,A:NEXT:SYS700
   :DATA120,169,201,141,20,3,169,2
30 DATA141,21,3,88,96,165,197,201,4,208,3
   ,238,32,208,76,49,234
```

Voor wie het nog niet wist, voegt Job er nog de volgende POKEs aan toe:

POKE198,0:WAIT198,1	- Wacht tot er een toets wordt ingedrukt.
POKE788,80	---------- Tekenen met de cursortoetsen.
POKE53265,80	- Rode cursor en <Shift+Letters> witte achtergrond.

Loeisnel

Een loeisnelle priemgetallengenerator kwam van Jeroen Hermkens uit Scherpenzeel. Hij vond een nieuwe algoritme om priemgetallen (getallen dus, die alleen deelbaar zijn door 1 en zichzelf) te berekenen tot de computer erbij neervalt.

```
1 A=5:A=A+B:FORI=3TOSQR(A)STEP2:IFA/I=INT(A/I)
  THENB=B+2:GOTO1
2 NEXT:PRINTA,:B=B+2:GOTO1
```

Belgische vlag

Voor de C128 schreef Raoul Publie uit Willebroek in België een grafisch programma, waaruit iets van vaderlandsliefde spreekt.

```
40 GRAPHICS3,1:COLOR0,2:COLOR1,1:COLOR2,8:COLOR3,3
   :COLOR4,2:WIDTH1:C=1
50 FORJ=30TO100STEP35
60 BOXC,J,15,J+35,180,,1:C=C+1
70 NEXT
80 X=54:Y=34
90 FORJ=1TO40:READXN,YN:DRAW0,X,YTOXN,YN:X=XN:Y-
   =YN:NEXT
100 DRAW0,X,YTO54,34
110 PAINT0,82,24,0
120 GETKEYA$:GRAPHIC0:END
130 DATA57,44,67,47,82,25,82,30,84,30,88,25,89,26,89,32
    ,92,32,94,27,96,28,97,39
140 DATA100,43,107,41,107,47,113,50,109,77,121,82,125,90
    ,121,96,126,100,128,110
150 DATA121,126,118,121,112,144,116,165,108,172,90,150,90
    ,125,88,125,84,139
160 DATA73,140,72,114,61,111,58,96,54,100,48,77,41,84,34
    ,75,33,54
```

Cirkels

Blijven we nog even bij de C128 dan komen we bij Ritchie Zebeda uit Rotterdam, die al met vijf dagen programmeerervaring dit miniatuurtje tevoorschijn toverde.

```
10 GRAPHIC1,1
20 FORA=0TO100STEP5
30 CIRCLE 1,A,A,A,A
40 NEXTA
50 CHAR 1,25,1,"COMMODORE INFO"
60 CHAR1,25,2,"<14 X SHIFT + C>"
```

En dan nog maar een:

```
10 GRAPHIC 1,1
20 FORA=0TO199STEP5
30 DRAW 1,A,0 TO 200-A,199
40 DRAW 1,0,A TO 200,199-A
50 NEXTA
60 FORA=0TO25STEP2
70 CHAR 1,27,A,"COMMODORE 128"
80 NEXT
```

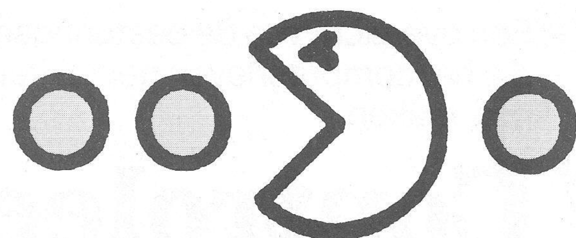
Pokes

Het blijft Pokes regenen op mijn bureau, zoals deze van Marco Lens uit Rotterdam.

POKE 792,252:POKE 793,252 - Deze POKE verandert de RESTORE-toets in een reset-toets.

POKE 120,0 - Verandert de C64 in een tekstverwerker. Basic is uitgeschakeld.

iatuurtjes



‹orte tot zeer korte programma's.

POKE 2050,1:SYS 42291:LIST - Geeft een Basic-programma terug na NEW.

Commodore 16

Nog steeds klachten over een gebrek aan belangstelling in Miniatuurtjes voor de C16. Met enige handigheid kan dit programma van Ronald Molenaar uit Castricum ook voor de C64 geschikt worden gemaakt.

```
5 COLOR4,1:COLOR0,1:COLOR1,2
10 FORZ=64TO127
20 PRINT"<CLR/HOME>":PRINT"<HOME> <12 X NEER 12 X
   RECHTS> <RVS ON> LEES DE C=INFO!!<RVS OFF>"
30 A=3073:B=3110:C=3111:D=4031:E=4071:F=4033
   :G=4032:H=3072
40 FORU=1TO11
50 FORI=ATOB:POKEI,Z:NEXTI
60 FORJ=CTODSTEP40:POKEJ,Z:NEXTJ
70 FORK=ETOFSTEP-1:POKEK,Z:NEXTK
80 FORL=GTOHSTEP-40:POKEL,Z:NEXTL
90 A=A+40:B=B+40:C=C-1:D=D-1:E=E-40:F=F-40
   :G=G+1:H=H+1:NEXTU:NEXTZ
```

Cijferorgel

Maar laten we de C64 niet vergeten. Voor deze populaire computer schreef Sietse Witzenburg uit Bergum in Friesland een orgel, dat met de cijfertoetsen wordt bediend.

```
10 S=54272:FORT=0TO24:READA:POKES+T,A:NEXT
20 GETA$:IFA$=""THEN20
30 C=VAL(A$)+2:IFC=2THENC=12
35 H=C+5*C
40 POKES+4,64:POKES+11,64:POKES+18,64
50 POKES+1,C:POKES+8,H:POKES+15,H
60 POKES+4,65:POKES+11,65:POKES+18,65
80 GOTO 20
100  DATA,,200,200,,26,,,,200,200,,26,,9,,200,200,,26,,,,,15
```

STER

Nog een keer de STER-jingle voor de C64:

```
10 S=54272:POKES+24,15:POKES+5,10
   :POKES+6,90:POKES+3,45:POKES+4,65:READT
20 IFT=-1THENPOKES+4,64:END
30 IFT=1THENPOKES+4,64:FORI=1TO800:NEXT
40 IFT=0THENFORI=1TO120:NEXT:GOTO10
50 POKES+1,T:FORI=1TO150:NEXT:POKES+4,64:GOTO10
60 DATA126,119,126,100,75,75,75,84,100,112,126,1
   ,126,126,134,150,126,100,84
70 DATA 100,0,112,100,0,200,-1
```

Simon

Veel inzendingen voor Simon's Basic, dat de mogelijkheden van de C64 toch wel een stuk vergroot. Drie grafische miniaturen van Phlippe Leybaert uit Zelzate, België. Tussen haakjes '<PI>' in de listings is Shift grote pijl omhoog op het C64-toetsenbord.

```
10 HIRES1,0
20 FORT=1TO4*PI STEP PI/40
30 S=SIN(T):LINE 0,100,T/PI*80,100+S*80,1
40 NEXT
50 GOTO 50
```

```
10 HIRES1,0
20 FORT=0TO4*<PI> STEP <PI>/40
30 S=SIN(T):LINE 160,100,80*T/<PI>,100+S*80,1
40 NEXT
50 GOTO 50
```

```
10 HIRES1,0
20 X2=200
30 XA=5:YB=5
40 LINE X1+60,Y1,X2+60,Y2,1
50 X1=X1+XA:Y1=Y1+YA:X2=X2+XB:Y2=Y2+YB
60 IF X1=200 AND Y1=0 THEN XA=0:YB=0:YA=5:XB=-5
70 IF X1=200 AND Y=200 THEN XA=-5:YB=-5:YA=0:XB=0
80 IF X1=0 AND Y1=200 THEN XA=0:YB=0:YA=-5:XB=5
90 IF X1=0 AND Y1=0 THEN 90
100 GOTO 40
```

Ook voor Simon's Basic van Donald Boshom uit Rotterdam:

```
10 HIRES1,0
20 FORX=0TO320STEP5
30 X1=X*5/7.5:X2=X/5
40 M1=175/329*X
50 LINE320-X1,M1,75+X1,75,1
60 NEXTX
70 FORY=0TO50STEP5
80 CIRCLE100,100,Y,10,1
90 NEXTY
100 WAIT298,255
```

En van Albert Edens uit Haren:

```
5 PRINT CHR$(147)
10 HIRES1,0
15 BFLASH 15,0,1
20 FORX=1TO360STEP4
30 ANGL 160,100,X,40*3.5,40,1
40 NEXTX
50 PRINTCHR$(147):POKE53281,1:PRINT"LEES COMMODORE
   INFO"
60 GOTO10
```

Ik kan er niets aan doen, maar lang niet alle inzendingen konden op deze twee pagina's worden geperst. We houden nog een flinke appel voor de dorst over. Inzendingen blijven overigens welkom vooral omdat kwaliteit in de Miniatuurtjes altijd voorrang krijgt. Veel programmeergenoegen.

Een overzicht van de besturingsmiddelen, die het computerleven gemakkelijker kunnen maken.

Controleer uw software

Spelbesturing in vele varianten

Ongetwijfeld is de joystick het meest bekende (en mishandelde) computeraccessoires. Behalve de vele typen joysticks bestaan er echter ook nog diverse andere "controls". Bijvoorbeeld trackballs, muizen en tekenpennen. In dit artikel gaan wij nader in op gewone en minder gewone controls.

De joystick

Joysticks waren in de eerste plaats bedoeld voor het spelen van videospelletjes. Daar hebben zij het qua gebruik en snelheidseisen dan `ook bijzonder hard te voortduren. Maar behalve voor het spelen van actiegames worden de "spelpookjes" eveneens gebruikt voor het tekenen op het beeldscherm, tekstbewerkingen, menukeuze en zelfs het besturen van robots. Hetgeen aanvankelijk voor uitsluitend voor vermaaksdoeleinden bestemd was heeft zich nu ontwikkeld tot een onmisbaar stuk randapparatuur. Niettemin zijn spelletjes de joysticktoepassing no 1. De C-64 en C-128 bezitten zoals bekend twee negenpolige D-connectors. Voor de meeste spelletjes met een pook moet u de joystick in port 2 (u mag ook poort zeggen) pluggen. Er zijn echter enkele dwarsliggers die port 1 voor de aansturing gebruiken. Denk bij een wijgering daarom niet direkt dat er iets met de apparatuur of het programma loos is, maar plug de stick eerst eens in de andere joystickport! De eigenlijke pook van de joystick kan men in het algemeen in vier richtingen bewegen. (De diagonalen zijn richtingkombinaties, al zijn sommige duurdere joysticks daarvoor met speciale sensors uitgerust). Vooruit, achteruit links en rechts. Als u de stick een bepaalde richting uitbeweegt wordt via een schakelaartje of een kontakt de stroomkring over het joystickcircuit tegen de massa kortgesloten. De computer registreert deze kortsluiting feilloos en past de spritebeweging overeenkomstig de stuutbevelen aan. Zoals gezegd bezitten de aansluitbussen 9 polen. Een is er voor de stroom aanvoer, 4 zijn er voor de beweging, en twee beschikbaar voor de vuurknoppen. De overblijvende pen blijft optioneel.

Aan de schakelaars of kontakten worden hoge eisen qua duurzaamheid, betrouwbaarheid, effectiviteit en snelheid gesteld. In depraktijk blijken de microschakelaars en de metaaltong (veer/drukpunt)- contacten het beste te voldoen. Nieuw zijn de kogelschakelaars. De pook is met een robuuste metalen kogel verbonden die tijdens de beweging tegen het vierwandige metalen omhulsel gedrukt wordt. Verder bestaan er nog kwikschakelaars waarbij kontakten door in de vier buisjes bewegende kwikdruppeltjes worden gesloten. Daanaast werken nog een aantal poken met lineaire potentiometer-schakelingen. Bij sommige spelen zijn zeer snel reagerende joysticks noodzakelijk. Anders kan je gewoonweg niet winnen!Bij de meeste Arcadespelletjes voldoet de doorsnee spelpook uitstekend. Een andere belangrijk punt betreft de handligging van de joystick. Dat is overigens een vrij persoonlijke kwestie. Alvorens een pook aan te schaffen is het raadzaam er eerst eens een half uurtje mee te spelen. Nog geen kramp of vermoeidheidsverschijnselen? Dan is dit wellicht de ideaal liggende spelpook. Let verder op de bereikbaarheid van de vuurknop en op het "rotsvast" staan van hetvoetstuk. De meeste sticks beschikken over rubberen anti-slipvoetjes. Andere over een zuignap.



Een bijzondere joystick is de joyball.

Nog even aandacht voor de kabelverbinding. Stevige en voldoend lange kabeltjes zijn een must, maar mogen niet door hun stijfheid in de weg zitten. Draadloze infrarood- besturing komt langzaam in maar vertoont nog veel mankementen.

De verschillende controltypen

Men kan joysticks en controls op verschillende manieren onderverdelen. Bijvoorbeeld naar het het aantal stuurrichtingen, vier of acht. Stuurknuppels met maar vier schakelrichtingen krijgen wat moeilijkheden tijdens het maken van diagonale (teken- en geavanceerde arcade-) bewegingen. Daarvoor moeten twee kontakten tegelijkertijd geactiveerd worden. Een aantal fabrikanten heeft dat goed opgelost, maar even uitproberen voor de aankoop is raadzaam. Andere joystickleveranciers gebruiken 8 schakelkontakten voor om de diagonale bestuurbaarheid te optimaliseren.

Een duidelijke onderverdeling is die in **"STICKS"**, **"BALLS"**, **"PENS"**, en **"MOUSES"**, oftewel poken, (stuur-)ballen, (licht-)pennen en muizen.

De sticks

De sticks zijn de bekende spelpookjes met een of meer vuurknoppen. Daarin is danook een bijzonder grote keuze. De vormgeving kan kwa doelstelling enigzins varieren. Zo lijkt de "COBRA" veel op de stuurknuppel van een vliegtuig en functioneert deze stick optimaal bij vluchtsimulatoren en niet al te snelle vliegspelletjes. De knuppel staat als een huis, maar is net zo traag als zijn echte vluigtuigcollega.

Een voorbeeld van een kogelschakelaarstick is de Totally-Accurate-Controller of TAC2. Daar de schakelweg zeer kort is reageert de stick, zoals diens naam belooft, bijzonder snel en accuraat. Ook een

korte slag heeft de van microschakelaars voorziene "Arcade Turbo" van SUZO. Deze is zowel voor rechts- als linkshandig gebruik geschikt. Verder zijn de kontakten goed tegen verontreinigingen (zenuwachtig roken, chips eten e. d.) beschermd en beschikt de stick over twee trefzekere vuurkonoppen. Een op de pook. De ander vóór in de driehoekige voetplaat. Zij zijn parallel of los te bedienen.
De joysticks krijgen steeds meer mogelijkheden. Sommigen hebben zelfs al veel weg van een druktoetsen- telefoonhoorn. Bijvoorbeeld de Power-Stick van de voormalige AMIGA Corporation (nu Commodore) met acht druppelkontakten en toetsenbordje voor Intellivision- e.a. geavanceerde spelen.

Trackballs

De bal is rond en zo ook de de "pook" van een Ball-Stick. I.p.v. aan het bekende knuppeltje draait de gebruiker aan een ronde stuurbal. Een groot voordeel van de Trackball is diens grote nauwkeurigheid bij moeilijke schermmanouvres. Er zijn bijvoorbeeld scherpe hoeken, de joystick komt maar tot 45 graden, mogelijk. Je kan de draaibal als een soort omgekeerde muis zien. De muis wordt echter met de arm verplaats terwijl de hand vrij blijft voor de functieknoppen. Bij de bal moet je het allemaal met de hand doen en zitten de knopjes aan de andere kant.
De toepassingen van de Trackball liggen meer op het grafische- dan spelletjes- terrein. Anders zijn er nauwelijks voordelen boven de joystick, terwijl de Trackball niet goedkoop is en extra software nodig heeft. In de grafische mode toont de draaibal zijn ware schildertalenten en is zijn prijs ten volle waard. Op beperkte schaal zijn bij sommige modellen zelfs CAD/CAM-achtige technieken, binnen de beperkingen van uw micro, mogelijk.

Allerlei mogelijke dikke/dunne lijnen, circels, boxen, fill, tekst&beeldkombinaties, fill, merge, load, save en print (op grafische!) afdrukkers liggen binnen het bereik van de Trackbal software. Momenteel zijn er al meerdere draaiballen leverbaar: De Marconi RB2 (C-64), Wico's Trackball voor de ATARI videocomputer, Zircon's Trackball Controller met twee vuurknoppen en automaties vuren, Wico- en Atari- Trackball.

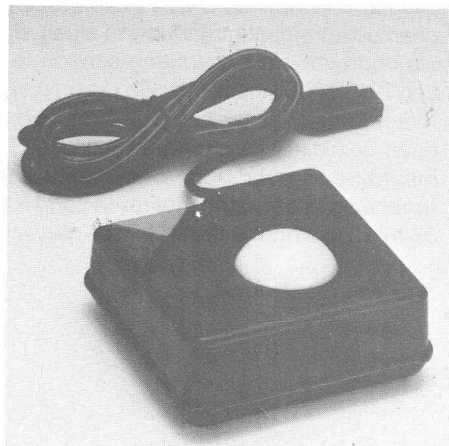
Light-Pens

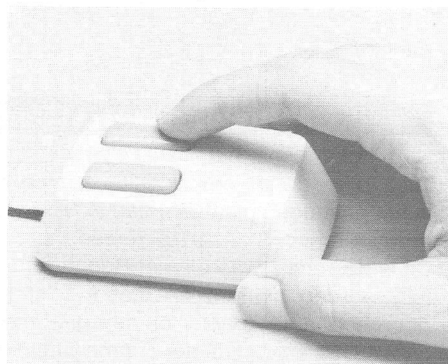
Lichtpennen zijn de grafische ontwerpcontrols bij uitstek. Het aantal verkrijbare pennen is niet zo groot, maar er zijn vele eenvoudige zelfbouwschema's in omloop. Ook voor de lichtpennen is additionele software noodzakelijk anders gebeurt er vanzelfsprekend niets. Bij een aantal tekenprogramma's is overigens al in het gebruik van een lichtpen voorzien.

Een van de bekendste grafische "pennen" of "pads" is de **Koala Pad** van Koala Technologies voor de C-64. Apple en Atari. Het eigenlijke "Pad" dient als "schetsboek" en toetsenbord. Eventueel is het Pad ook als controlepaneel voor stragieen simulatiespelen bruikbaar. I.p.v. een echte pen is gewoon de eigen vinger of een griffel voldoende om figuren, shaduwen of penseelstreken op het beeldscherm aan te brengen. Het gaat dus eigenlijk meer om een lightpad dan een lightpen.
In samenwerking met compatibele software zijn met de pen ware meesterwerken aan computerkunst te scheppen. Een extra optie is het gebruik van de Koala Pad als (semi-)intelligent toetsenbord voor zakelijke- en financiële programma's. Met de bijpassende overlays zijn zo'n 40 functietoetsen te definiëren.
Sinds kort is tevens een echte Koala lichtpen verkrijgbaar. Samen met de PENPAINTER software kunt u de grafische voorstellingen pixel voor pixel(!) opbouwen.
Een andere lichtgriffel komt van Lindy. De lichtpenset bestaat uit een insteekmoduul (C-64) en lichtpen met portstecker. Een vreemde eend in de pennenbijt is de "SPACEPEN" met 3-dimensionale besturing van Soniture USA. De SPACEPEN werkt niet alleen in een vlak (op de x- en y-as), maar kan ook in de diepte (de z-as) werken. Zo kunnen grafische voorstellingen niet alleen naast-, maar ook achter elkaar getekend worden. Indien de 3D-weergave niet nodig is zet u gewoon de D-schakelaar in de 2D-positie.
De commando-overdracht verloopt m.b.v. ultrasoongeluid. Daarvoor is uiteraard een speciale computerontvanger, het receiver frame, nodig. De werkruimte voor deze receiver bedraagt een krappe twee meter. Behalve voor grafische toepassingen is de Spacepen ook ideaal voor gehandicapten, videospelletjes, robotcontrole, industriele toepassingen en numerieke-controle. De Space pan kost c. a. fl 500, -, is met de meeste grafische software compatibel en past op de C-64, ATARI, MSX, APLLE en IBM - computers.

Muizen

Deze electronische knaagdieren zijn genoegzaam bekend van de gebruiksvriendelijke (menu-)besturingssystemen op professionele micro's als de MAC, ATARI 520 ST en de AMIGA. Voor de huiscomputers waren er tot voor kort geen muizen. Onlangs kwamen er echter enige types voor de C-64 op de markt.
Het Britse bedrijf SMC brengt in haar "Connections"-lijn nu de "Magic Mouse". Deze muis is primair voor grafische doeleinden ontworpen, maar staat ook zijn/haar (?) mannetje als joystick. Met deze muis kunt u zes grafische voorstellingen in kleur naar de disk saven. Verder kunnen er allerlei grafische objecten zoals frame boxes, circels, line rays, horizontale- en vertikale lijnen en 200 sprites mee ontworpen worden. Ook het vervaardigen van ikonen, inkleuren en tekstbewerkingen behoren tot de mogelijkheden. Ideaal voor de muiscompatibele (!) softwarepakketten bij een relatief lage prijs van circa *f* 250,-.





Er zijn nogal wat "controls" verkrijgbaar met elk hun optimale toepassingsgebied. Sommige zijn primair voor spellen, andere meer voor grafische toepassingen bedoeld. Een aantal controls weert zich aardig op beide terreinen, maar u zult toch een compromis moeten accepteren, tenzij u de hele zaak tegelijk wilt aansluiten op uw machine.

Print-out

Met o.a. Boem 64 - Toupen - Morse - Kop van Jut - Morse C-16 - Kleinkunst C-128

Konden we in het vorige nummer de C-16 rubriek introduceren, deze keer is het de beurt aan de **C-128**. Wel moet U de listings voor de C-128 wel met extra grote zorgvuldigheid intikken, omdat er nog geen aangepaste versie van de Checksum- en Visilistprogramma's beschikbaar is. We hopen die uiteraard zo snel mogelijk klaar te hebben.
Het aantal ingestuurde 128-programma's is tot nu toe nog gering, dus 128-bezitters, we hopen nog van jullie te horen. Als U een programma voor de C-128 instuurt, zorg er dan wel voor dat er geen besturingscommando's in staan, dus bv. een **inverse hartje** als opdracht om het scherm schoon te maken vervangen door een CHR$ (147) instructie. Op die manier wordt voorkomen dat er fouten ontstaaan bij het afdrukken van de listing zolang er nog geen geschikt Visilistprogramma is.
Inlichtingen over door ons geplaatste listings kunt U elke maandag telefonisch inwinnen van 16.00 tot 21.00 uur, 02152 - 63431. Op andere tijden is dat niet meer mogelijk.

Rob van den Heuvel.

Inhoud van dit listingdeel

Checksum-listtester 64

Nog maar eens het Syntax-checksum programma om uw intikwerk te controleren. Intikken, saven, en dan via RUN onderin in het geheugen plaatsen, dan te testen programma laden (niet runnen) en via SYS 49152 testen.
Dit is de versie, waarbij met Run/stop de listing op het scherm kan worden gestopt en met F-1 of F-7 weer kan worden doorgegaan. Op papier printen gaat met : OPEN 4,4: CMD 4: SYS 49152 en dan na het printen weer PRINT #4: CLOSE 4 om het af te sluiten. Niet alle fouten haalt u hiermee uit uw programma's, want omwisselingen zoals uv ipv. vu komen er niet uit, maar vooral dataregels kunt u er heel goed mee controleren.

```
  1 rem ****************************************
    *****************
  2 rem              basic loader   "SYNTAX.CHECKS
    UM"
  3 rem  runnen en new.  blijft in ram op $c000.
  4 rem          checksum testen met commando  :
    sys 49152'
  5 rem
  6 rem v.840928.64    jan bodzinga
  7 rem ****************************************
    *****************
 10 i=49152  :rem beginadres
 20 reada:ifa<0then50  :rem data ingelezen
 30 pokei,a:i=i+1:goto20
 50 poke49184,148:poke49185,192
 55 i=49300
 60 read a: ifa<0then90
 70 pokei,a:i=i+1:goto60
 90 print"data is weggezet"
 95 print"checksum testen met sys49152"
100 data 165,  43, 166,  44, 133, 163, 134, 164,
    169, 147
110 data  32, 210, 255, 160,   0, 240,   3,  32,
     73, 192
120 data  32,  73, 192, 208,   1,  96,  32, 225,
    255, 208
130 data   3,  76, 116, 164,  32,  81, 192,  32,
     73, 192
140 data 240,  12, 201,  32, 240, 247,  24, 101,
    167, 133
150 data 167,  76,  37, 192, 166, 167, 169,   0,
    132, 168
160 data  32, 205, 189, 169,  13,  32, 210, 255,
    164, 168
170 data  76,  17, 192, 200, 208,   2, 230, 164,
    177, 163
180 data  96, 162,   0, 189, 123, 192, 240,   6,
     32, 210
190 data 255, 232, 208, 245,  32,  73, 192, 170,
     32,  73
200 data192, 132, 168,  32, 205, 189, 162,   3, 1
    69,  32
210 data  32, 210, 255, 202, 208, 250, 169,   0,
    133, 167
220 data 164, 168,  96,  82,  69,  71,  69,  76,
     32,   0
230 data  -1
240 data 165,197,201,3,240,7,201,4,240,6
250 data 76,148,192,76,34,192,169,147,32,210
260 data 255,76,161,192
270 data -1

            *** EINDE LISTING ***

Checksum: Checksum.nw

       regel   1     249       regel  110    158
       regel   2     84        regel  120    232
       regel   3     6         regel  130    183
       regel   4     117       regel  140    96
       regel   5     143       regel  150    96
       regel   6     17        regel  160    127
       regel   7     61        regel  170    71
       regel  10      157      regel  180    223
       regel  20      65       regel  190    73
       regel  30      203      regel  200    79
       regel  50      14       regel  210    109
       regel  55      251      regel  220    106
       regel  60      193      regel  230    225
       regel  70      207      regel  240    114
       regel  90      245      regel  250    89
       regel  95      237      regel  260    68
       regel  100      183     regel  270    225
```

UTILITIES 1

Smooth Screen Clear

Wie veel met teksten werkt tijdens het programmeren, zal gemerkt hebben, dat er af en toe een schoon scherm nodig is, wil je het scherm tenminste niet vol laten lopen met tekst. Daarvoor heeft de Commodore natuurlijk een zeer handige en korte oplossing in de vorm van het inverse hartje, ofwel het symbool, dat verschijnt na het intypen van SHIFT/CLR-HOME. Wie niet van rare tekens op het scherm of papier houdt, gebruike Visilist of het commando PRINT CHR$(147) om het scherm schoon te maken. Daarbij is het meestal ook gewenst om wat ruimte bovenin het scherm te makend door na het hartje wat inverse Q's te zetten (CURSOR DOWN).
Maar veel fraaier en gelijkmatiger is onderstaande routine om het scherm op te schonen.

```
 1 rem smooth scrn clr  64
 2 rem door p bos
 3 rem meedhuizen / 05963-565
 4 rem
11 rem       ****** demo  *********
12 poke52,158:poke56,158
13 forx=0to223:reada:poke40448+x,a:next
14 sys40448
15 forx=1to200:a$=a$+chr$(int(rnd(1)
   *30)+65):next:poke53281,1:poke646
   ,0
16 printchr$(147):forx=1to4:printa$:
   next
17 sys 40611:forpa=1to800:next:sys40
   547:poke53281,1:forpa=1to800:next
   :goto17
18 rem p.s kleur veranderen d.m.v :
19 rem  poke40523,kl
20 rem  poke40548,(peek(40548)and240
   )+kl
50 data169,158,133,52,133,56,120,169
   ,37,141,20,3,169,158
51 data141,21,3,173,17,208,41,127,14
   1,17,208,169,148,141
52 data18,208,169,240,141,26,208,88,
   96,173,26,208,41,1
53 data240,7,173,25,208,41,1,208,3,7
   6,49,234,173,33
54 data208,41,15,240,13,169,0,141,33
   ,208,173,97,158,141
55 data18,208,208,11,169,1,141,33,20
   8,173,98,158,141,18
56 data208,173,25,208,141,25,208,104
   ,168,104,170,104,64,48
57 data250,169,145,205,18,208,208,25
   1,141,33,208,169,127,141
58 data13,220,169,148,141,97,158,169
   ,150,141,98,158,169,241
59 data141,26,208,206,97,158,238,98,
   158,162,248,160,0,200
60 data208,253,232,208,248,173,97,15
   8,201,48,208,233,169,240
61 data141,26,208,169,129,141,13,220
   ,96,169,251,205,18,208
62 data208,251,169,127,141,13,220,16
   9,48,141,97,158,169,250
63 data141,98,158,169,241,141,26,208
   ,238,97,158,206,98,158
64 data162,248,160,0,200,208,253,232
   ,208,248,173,97,158,201
65 data147,208,233,169,240,141,26,20
   8,169,129,141,13,220,96

*** EINDE LISTING ***
```

```
smooth screen clear

regel 1    234        regel 52   162
regel 2    247        regel 53   255
regel 3    128        regel 54   195
regel 4    143        regel 55   206
regel 11    42        regel 56   88
regel 12    206       regel 57   194
regel 13    248       regel 58   170
regel 14    162       regel 59   57
regel 15    139       regel 60   210
regel 16    15        regel 61   98
regel 17    32        regel 62   160
regel 18    74        regel 63   133
regel 19    127       regel 64   142
regel 20    230       regel 65   148
regel 50    3
regel 51    242
                      ready.
```

Boem 64

Mocht u graag 'knallende' programma's maken en heeft u problemen met het in geluid omzetten van uw explosieve gedachten, geen nood. Uit Gent ontvingen wij de volgende routine, deze kunt u aan uw eigen programma's koppelen... et voila!

```
 1 rem boem
 2 rem door k. somerling
 3 rem gent belgie
 4 rem
10 a=54273:pokea+23,15:pokea+5,0:poke
   a+4,12:pokea,20:pokea+3,129:forb=
   1to2300
20 nextb:pokea+3,0:end

*** EINDE LISTING ***
```

```
boem 64

regel 1    178        regel 10   23
regel 2    236        regel 20   201
regel 3    101
regel 4    143        ready.
```

UTILITIES 2

Dataregelaar

Machinetaal kan in de vorm van dataregels in een Basic programma worden ingevoegd. Als u een stukje machinetaal in dataregels wilt omzetten kunt u van dit programma gebruik maken. Na het intikken van het start- en eindadres, het regelnummer van de basicloader en de dataregels maakt het programma de dataregels aan. Ook de optie, die dit programma biedt, om dataregels te verwijderen kan van pas komen.

```
    1 rem dataregelaar 1
    2 rem door marc de hingh
    3 rem uit breda / 076-145652
    4 rem
59996 rem - run 60000 om een geheugenbl
      ok naar data te schrijven
59997 rem - run 60020 om een gedeelte v
      an het programma te verwijderen
59998 rem
59999 stop
60000 input"[CLR-HOME]startadres";s:a=s
      :input" eindadres";e:input"data v
      anaf regel";r
60001 print"max. lengte dataregels (<79
      )":inputk
60002 input"basicloader op regel";l:poke
      53265,11
60003 print"[CLR-HOME][neer]":n=0
60004 printr"[links]data";
60005 printmid$(str$(peek(a)),2);:a=a+1
      :ifa>ethen60009
60006 ifpos(0)<=k-len(str$(peek(a)))then
      print",";:goto60005
60007 print:r=r+1:n=n+1:ifn<9then60004
60008 print"r="r":a="a":s="s":e="e":k="
      k":l="l":goto60003":goto60012
60009 print:print"s="s":e="e":l="l":got
      o60010":goto60012
60010 print"[CLR-HOME][2xneer]"l"[links]
      fOa="mid$(str$(s),2)"to"mid$(str$
      (e),2)":rEx:pOa,x:nE"
60011 print"poke198,0:sys65126"
60012 poke198,10:fori=0to9:poke631+i,13
      :next:print"[HOME]";:end
60020 print"[CLR-HOME]regels verwijdere
      n":input"eerste[shift-SPATIE]rege
      l";r:input"laatste regel";rr
60021 poke53265,11
60022 print"[CLR-HOME][neer]":n=0
60023 printr:r=r+1:n=n+1:ifr>rrthen6001
      1
60024 ifn<9then60023
60025 print"r="r":rr="rr":goto60022":goto
      60012
```

*** EINDE LISTING ***

dataregelaar

regel 1	29	regel 60005	161
regel 2	221	regel 60006	70
regel 3	15	regel 60007	7
regel 4	143	regel 60008	177
regel 59996	12	regel 60009	149
regel 59997	166	regel 60010	69
regel 59998	143	regel 60011	71
regel 59999	144	regel 60012	82
regel 60000	254	regel 60020	61
regel 60001	147	regel 60021	42
regel 60002	219	regel 60022	235
regel 60003	235	regel 60023	196
regel 60004	33	regel 60024	103
		regel 60025	46

ready.

Dataregelaar 2

Dit programma is vrijwel hetzelfde als **dataregelaar**. Het verschil zit hem in het feit dat de dataregels compacter worden gemaakt, hetgeen ten koste van de overzichtelijkheid gaat. Op zich is dat geen probleem, het is echter niet prettig om een listing te moeten overtikken die een lap **'compacte data'** bevat. Bij het gebruik van dit programma zult u dat dus wel even in overweging moeten nemen.

```
    1 rem dataregelaar 2
    2 rem door marc de hingh
    3 rem uit breda / 076-145652
    4 rem
59996 rem - run 60000 om een geheugenb
      lok naar data te schrijven
59997 rem - run 60020 om een gedeelte
      van het programma te verwijderen
59998 rem
59999 stop
60000 input"[CLR-HOME]startadres";s:a=
      s:input" eindadres";e:input"data
       vanaf regel";r
60001 print"max. lengte dataregels (<7
      9)":inputk
60002 input"basicloader op regel";l:poke
      53265,11
60003 print"[CLR-HOME][neer]":n=0
60004 printr"[links]data ";
60005 d=peek(a):d1=int(d/10):d2=d-10*d
      1:printchr$(d1+65)chr$(d2+48);
60006 a=a+1:ifa>ethen60010
60007 ifpos(0)<k-1goto60005
60008 print:r=r+1:n=n+1:ifn<9then60004
60009 print"r="r":a="a":s="s":e="e":k="
      k":l="l":goto60003":goto60015
60010 print:print"s="s":e="e":l="l":go
      to60011":goto60015
60011 print"[CLR-HOME][2xneer]"l"a="mid$
      (str$(s),2)
60012 printl+1"[links]rEx$:fOi=1tolen(
      x$)/2:x=10*(aS(mI(x$,i*2-1))-65)
      +aS(mI(x$,i*2))-48
60013 printl+2"pOa,x:a=a+1:nE:ifa<"mid$
      (str$(e+1),2)"goto"mid$(str$(l+1),2)
60014 print"poke198,0:sys65126"
60015 poke198,10:fori=0to9:poke631+i,1
      3:next:print"[HOME]";:end
60020 print"[CLR-HOME]regels verwijder
      en":input"eerste[shift-SPATIE]re
      gel";r:input"laatste regel";rr
60021 poke53265,11
60022 print"[CLR-HOME][neer]":n=0
60023 printr:r=r+1:n=n+1:ifr>rrthen60014
60024 ifn<9then60023
60025 print"r="r":rr="rr":goto60022":goto
      60015
```

*** EINDE LISTING ***

dataregelaar 2

regel 1	30	regel 60002	219	regel 60012	95
regel 2	221	regel 60003	235	regel 60013	177
regel 3	15	regel 60004	33	regel 60014	71
regel 4	143	regel 60005	42	regel 60015	82
regel 59996	12	regel 60006	169	regel 60020	61
regel 59997	166	regel 60007	35	regel 60021	42
regel 59998	143	regel 60008	7	regel 60022	235
regel 59999	144	regel 60009	180	regel 60023	199
regel 60000	254	regel 60010	153	regel 60024	103
regel 60001	147	regel 60011	129	regel 60025	49

ready.

Toupen

Naast schaakcomputers kunt u tegenwoordig ook computers, die het bridgespel beheersen, in speelgoedwinkels aantreffen. Als homecomputer bezitter kunt u natuurlijk ook enkel een stukje software kopen en hiermee uw elektronische huisgenoot tot grootmeester of bridgevedette omtoveren. Deze software kost dan wel aanmerkelijk minder dan een complete computer, maar het blijft een flinke duit. In principe kan uw computer in bijna elk spel tot een geducht tegenstander worden. Het enige dat u hiervoor nodig hebt is de juiste programmatuur. Helaas komt er voor het maken van een dergelijk programma nogal wat kijken. Inmiddels zijn er voor een aantal kaartspelen dergelijke programma's geschreven en op de markt gebracht. U kunt nu dus poker, bridge of bijvoorbeeld het spel pesten met uw computer spelen. Tot ons plezier kunnen wij u het spel toupen aanbieden. Dit leuke spel waren wij tot nog toe niet tegen gekomen. Het is spel is erg bekend en als u de spelregels niet kent dan is er vast wel iemand in uw vriendenkring die u die kan uitleggen.

```
 10 rem toepen / cmb-64
 20 rem door r.a. wolswijk
 30 rem zeist / 03404-23295
 40 rem
 50 print"[CLR-HOME]":dim krt(32),krt
    $(32):slag$="s":b=214:o=58732
 60 v=54296:a=54277:r=54278:fh=54273:
    fl=54272:g=54276
 70 poke 53280,15:poke53281,15:print"
    [ZWART]"
 80 naam$="":input"[CLR-HOME]tik je n
    aam in";naam$:if naam$="" or len(
    naam$)>8 then 80
 90 gosub 100:goto 120
100 print"[HOME]UCCCCCCCCCCI":print"B
    *toepen* B":print"B(c) c-infoB"
110 print"B[2xspatie]1986[4xspatie]B"
    :print"JCCCCCCCCCCK":return
120 a$="":print"[HOME]";:for wis=0 to
     14 :printtab(12)"[13xspatie]":next
130 pokeb,8:syso:for wis=8 to 10:print
    "[13xspatie]":next
140 for wis=1 to 4:ckleur$(wis)="":sk
    leur$(wis)="":next
150 restore:cruil=0:sruil=0:ct=0:gd=0
    :i=0:gosub 2440
160 rem ** kaarten inlezen **
170 for p=1 to 32:read krt(p),krt$(p)
    :next
180 pokeb,23:syso:printtab(14)"[12xspatie
    ]"
190 pokeb,15:syso:for wis=15 to 22
200 printtab(5)"[32xspatie]":next
210 rem ** var.omzetting v.printrouti
    ne **
220 x=5:y=15:cklop=1:slag=0:keer=0:kk
    eer=0:s=0:c=0:gosub 1450
230 for p=1 to 4:gosub 2400
240 if spkrt(p)>4 then 300
250 if spkrt(p)=1 then tal$(p)="b"
260 if spkrt(p)=2 then tal$(p)="v"
270 if spkrt(p)=3 then tal$(p)="h"
280 if spkrt(p)=4 then tal$(p)="a"
290 goto 330
300 if spkrt(p)=10 then tal$(p)="10":
    goto 330
310 tal$(p)=right$(str$(spkrt(p)),1)
320 rem ** kaarten uitdelen **
330 pokeb,17:syso:printtab(x+1)tal$(p
    );tab(x+4)spkrt$(p)
340 print"[neer]";tab(x+1)spkrt$(p);tab(
    x+4)spkrt$(p)
350 print"[neer]";tab(x+1)spkrt$(p);:
    if tal$(p)="10" then x=x-1
360 printtab(x+4)tal$(p):if tal$(p)="
    10" then x=x+1
370 x=x+8:next p:gosub 2630:if a$="n"
     then 120
380 if sruil=1 or zpunt=14 or cpunt=1
    4 then 500
390 pokeb,23:syso:printtab(14)"[RVS-aan]
    ruilen?(j/n)"
400 rem ** kaarten ruilen v.speler **
410 get r$:if r$="" then 410
420 if r$="n" then 500
430 if r$<>"j" then 410
440 print"[op][12xspatie]"
450 sruil=1:kijk=int(3*rnd(0)+1):if k
    ijk=2 then 180
460 pokeb,3:syso:printtab(15)"[RVS-aan]
    ik kijk!":gosub 2950:printtab(15)
    "[op][8xspatie]"
470 for k=1 to 4:if spkrt(k)<5 then 4
    90
480 pokeb,3:syso:printtab(15)"[RVS-aan]
    blufkont":zpunt=zpunt+1:gosub 295
    0:goto 180
490 next k:cpunt=cpunt+1:goto 180
500 pokeb,23:syso:printtab(14)"[15xspatie
    ]":gosub 1540
510 if slag$="s" then 1110
520 comkrt(c)=0:pokeb,23:syso:printtab(
    14)"mijn beurt!"
530 if ct>28 then gosub 1950
540 if a$="n" then 120
550 rem ** comp.zoekt 'n kaart **
560 c=int(4*rnd(0)+1):if comkrt(c)=0
    then 560
570 kslag=slag*3/2:if kslag<>int(slag
    *3/2) then 610
580 if slag>3 then gosub 1790:if a$="
    n" then 120
590 goto 800
600 rem ** kleur bekennen comp. **
610 for p=1 to 4:if comkrt(p)=0 then
    740
620 if comkrt$(p)<>spkrt$(s) then 740
630 rem ** indien kl.comp=kl.speler z
    oek gr./kl.kaart **
640 for l=1 to 4:if comkrt(l)=0 then
    700
650 if comkrt$(l)<>spkrt$(s) then 700
660 if comkrt(l)<spkrt(s) then 700
670 c=l:if slag=7 then gosub 1950
680 if slag>4 then gosub 1790:if a$="
    n" then 120
690 goto 800
700 next l:klein=comkrt(p):for k=1 to
     4:if comkrt(k)=0 then 730
710 if comkrt$(k)<>spkrt$(s) then 730
720 if comkrt(k)<=klein then klein=co
    mkrt(k):c=k
730 next k:goto 770
740 next p:gosub 2870:for p=1 to 4:if
     spkrt$(s)<>ckleur$(p) then 760
750 goto 780
760 next p
770 if slag=7 then gosub 1870:if a$="
    n" then 120
780 ckleur$(s)=spkrt$(s)
790 rem ** comp.kaart printen **
800 slag=slag+1:print"[HOME]";:for wi
    s=0 to 6:printtab(15)"[8xspatie]"
    :next
810 gosub 2750:print"[HOME]";:printtab(
    17)"UCCCCI":printtab(17)"B[4xspatie]
    B"
```

```
820 printtab(17)"B[4xspatie]B":printtab(
    17)"B[4xspatie]B":printtab(17)"B[4xsp
    atie]B"
830 printtab(17)"B[4xspatie]B":printtab(
    17)"JCCCCK"
840 pokeb,1:syso:printtab(18)ctal$(c)
    :pokeb,1:syso:printtab(21)comkrt$
    (c)
850 pokeb,3:syso:printtab(18)comkrt$(
    c)
860 x=21:pokeb,3:syso:printtab(21)com
    krt$(c):pokeb,5:syso
870 printtab(18)comkrt$(c):if ctal$(c
    )="10" then x=20
880 pokeb,5:syso:printtab(x)ctal$(c):
    pokeb,3:syso:printtab(31)slag
890 gosub 2940:ct=ct-comkrt(c):i=i+1:
    kaart(i)=comkrt(c):kaart$(i)=comk
    rt$(i)
900 kslag=slag*3/2:if kslag=int(slag*
    3/2) then 930
910 goto 1110
920 rem ** wie is aan slag **
930 if slag$="s" then 1020
940 if comkrt$(c)=spkrt$(s) then 970
950 if slag=8 then gosub 2950:gosub 2
    850:goto 120
960 slag$="c":goto 520
970 if comkrt(c)<spkrt(s) then 1000
980 slag$="c":if slag=8 then gosub 29
    50:gosub 2850:goto 120
990 goto 520
1000 slag$="s":if slag=8 then gosub 29
    50:gosub 2860:goto 120
1010 goto 1110
1020 if spkrt$(s)=comkrt$(c) then 1050
1030 if slag=8 then gosub 2950:gosub 2
    860:goto 120
1040 slag$="s":goto 1110
1050 if spkrt(s)<comkrt(c) then 1080
1060 slag$="s":if slag=8 then gosub 29
    50:gosub 2860:goto 120
1070 goto 1110
1080 slag$="c":if slag=8 then gosub 29
    50:gosub 2850:goto 120
1090 goto 520
1100 rem ** speler speelt 'n kaart **
1110 spkrt(s)=0:pokeb,23:syso:printtab(
    14)"[RVS-aan]jouw beurt!"
1120 get s$:if s$="" then 1120
1130 if s$="t" then 2050
1140 if s$="n" then slag$="s":zpunt=zp
    unt+cklop:goto 120
1150 s=val(s$)
1160 if spkrt(s)=0 then 1120
1170 if asc(s$)<49 or asc(s$)>52 then
    1120
1180 pokeb,10:syso:print"[10xspatie]":
    if slag>5 then 1260
1190 rem ** kleur bekennen speler **
1200 kslag=slag*3/2:if kslag=int(slag*
    3/2) then 1260
1210 for p=1 to 4:if spkrt$(p)<>comkrt
    $(c) then 1250
1220 if spkrt(p)=0 or comkrt(c)=0 then
     1250
1230 if spkrt$(s)<>comkrt$(c) then 112
    0
1240 goto 1260
1250 next p:skleur$(c)=comkrt$(c)
1260 if s$="1" then x=5
1270 if s$="2" then x=13
1280 if s$="3" then x=21
1290 if s$="4" then x=29
1300 rem ** kaart v. speler wissen **
1310 slag=slag+1:pokeb,15:syso:for wis
    =15 to 22:printtab(x)"[6xspatie]"
    :next
1320 rem ** kaart v.speler printen **
1330 pokeb,8:syso:for wis=8 to 14:print
    tab(17)"[7xspatie]":next
1340 pokeb,8:syso:printtab(17)"UCCCCI"
    :printtab(17)"B[4xspatie]B"
1350 printtab(17)"B[4xspatie]B":printtab(
    17)"B[4xspatie]B":printtab(17)"B[4xsp
    atie]B"
1360 printtab(17)"B[4xspatie]B":printtab(
    17)"JCCCCK"
1370 pokeb,9:syso:printtab(18)tal$(s):
    pokeb,9:syso:printtab(21)spkrt$(s
    )
1380 pokeb,11:syso:printtab(18)spkrt$(
    s):pokeb,11:syso
1390 printtab(21)spkrt$(s)
1400 x=21:pokeb,13:syso:printtab(18)sp
    krt$(s):if tal$(s)="10" then x=20
1410 pokeb,13:syso:printtab(x)tal$(s):
    pokeb,11:syso:printtab(31)slag
1420 gosub 2940:i=i+1:kaart(i)=spkrt(c
    ):kaart$(i)=spkrt$(c)
1430 kslag=slag*3/2:if kslag=int(slag*
    3/2) then 930
1440 goto 520
1450 rem ** 2x4 kaarten nemen **
1460 t=t+1:if t>4 then t=0:goto 1480
1470 gosub 1520:spkrt(t)=krt(z):spkrt$
    (t)=krt$(z):krt(z)=0:goto 1460
1480 if sruil=1 then  return
1490 t=t+1:if t>4 then t=0:return
1500 gosub 1520:comkrt(t)=krt(z):comkr
    t$(t)=krt$(z):krt(z)=0
1510 ct=ct+comkrt(t):goto 1490
1520 z=int(32*rnd(1)+1):if krt(z)=0 then
     1520
1530 return
1540 if cruil=1 or zpunt=14 or cpunt=1
    4 then return
1550 for p=1 to 4:if comkrt(p)>4 then
    1580
1560 next p:gd=1:goto 1610
1570 rem ** niet bluffen indien ct>18
    **
1580 if ct>18 then return
1590 rem ** bluffen v. 4 plaatjes **
1600 pl4=int(3*rnd(0)+1):if pl4=3 then
     return
1610 cruil=1:ct=0:pokeb,3:syso:printtab(
    12)"[RVS-aan]vier plaatjes! kijke
    n?(j/n)
1620 get k$:if k$="" then 1620
1630 if k$="n" then printtab(12)"[op][14xs
    patie]slag: 0[6xspatie]":goto 149
    0
1640 if k$<>"j" then 1620
1650 print"[HOME]";:for wis=0 to 4
1660 print"[39xspatie]":next
1670 rem ** kaarten v. comp. printen *
    *
1680 gosub 2750:x=5:y=0:for p=1 to 4:gosub
     2400
1690 pokeb,2:syso:printtab(x+1)ctal$(p
    );tab(x+4)comkrt$(p)
1700 print"[neer]";tab(x+1)comkrt$(p);
    tab(x+4)comkrt$(p)
1710 print"[neer]";tab(x+1)comkrt$(p);
    :if ctal$(p)="10" then x=x-1
1720 printtab(x+4)ctal$(p):if ctal$(p)
    ="10" then x=x+1
1730 x=x+8:next p:gosub 2950:print"[HOME]"
    ;:for wis=0to 7
1740 printtab(5)"[34xspatie]":next:gosub
     100
1750 rem ** stand na bluffen/terecht r
    uilen **
```

```
1760 if gd=1 then zpunt=zpunt+1:gd=0:gosub
      2630:goto 1490
1770 cpunt=cpunt+1:gosub 2630:goto 149
     0
1780 rem ** toepen v. comp. **
1790 if keer=1 then return
1800 if slag<>6 then 1840
1810 if comkrt(c)>7 then 1950
1820 for p=1 to 4:if comkrt$(c)=skleur
     $(p) then 1950
1830 next p
1840 if slag<2 and ct<18 then return
1850 if slag<6 and ct<6 then return
1860 if slag<8 and ct<5 then return
1870 if slag=7 and spkrt(s)=10 then return
1880 if slag<>7 then 1940
1890 for zz=1 to 4:if comkrt(zz)=0 then
      1910
1900 goto 1920
1910 next zz
1920 for z=1 to 7:if kaart(z)>spkrt(s)
      and kaart$(z)=spkrt$(s) then return
1930 next z
1940 tt=int(3*rnd(0)+1):if tt=2 then return
1950 if cpunt=14 or zpunt=14 or keer=1
      then return
1960 if cpunt+cklop+1>15 or zpunt+cklo
     p+1>15 then return
1970 cklop=cklop+1:gosub 2980:pokeb,8:
     syso
1980 print"toep voor";cklop:kkeer=0:ke
     er=1
1990 pokeb,10:syso:print"[RVS-aan]ga j
     e mee?(j/n)"
2000 get a$:if a$<>"j" and a$<>"n" then
      2000
2010 print"[op][15xspatie]"
2020 if a$="n" then zpunt=zpunt+cklop-
     1:slag$="s"
2030 return
2040 rem ** toepen v. speler **
2050 if kkeer=1 or zpunt=14 or cpunt=1
     4 then 1110
2060 if zpunt+cklop+1>15 or cpunt+cklo
     p+1>15 then 1110
2070 cklop=cklop+1:gosub 2980:pokeb,8:
     syso
2080 print"toep voor";cklop:keer=0:kke
     er=1
2090 kslag=slag*3/2:if kslag=int(slag*
     3/2) then 2120
2100 for p=1 to 4:if comkrt$(c)=skleur
     $(p) then 2360
2110 next p
2120 if slag<>6 then 2180
2130 for rr=1 to 4:if skleur$(rr)="" then
      2170
2140 for r=1 to 4:if comkrt(r)=0 then
     2160
2150 if comkrt$(r)=skleur$(rr) then233
     0
2160 next r
2170 next rr
2180 if slag<1 and ct>28 then 2360
2190 if slag<1 and ct<20 then 2330
2200 if slag<3 and ct<13 then 2330
2210 if slag<5 and ct<9 then 2330
2220 if slag<7 and ct<5 then 2330
2230 if slag<7 and ct>8 then 2360
2240 if slag=7 and comkrt(c)>9 then236
     0
2250 if slag=7 and comkrt(c)<3 then233
     0
2260 if slag<>7 then 2320
2270 for zz=1 to 4:if spkrt(zz)=0 then
      2290
2280 goto 2300
2290 next zz
2300 for z=1 to 7:if kaart(z)>comkrt(c
     ) and kaart$(z)=comkrt$(s) then 2
     330
2310 next z
2320 cc=int(3*rnd(0)+1):if cc<>2 then
     2360
2330 pokeb,10:syso:print"[RVS-aan]ik p
     as!":gosub 2950:cpunt=cpunt+cklop
     -1:slag$="c"
2340 if zpunt=14 then return
2350 goto 120
2360 pokeb,10:syso:print"[RVS-aan]ik g
     a mee!"
2370 if zpunt=14 then return
2380 goto 1110
2390 rem ** kaders v. kaarten printen
     **
2400 pokeb,y:syso:printtab(x)p:printtab(
     x)"UCCCCI":printtab(x)"B[4xspatie]
     B"
2410 printtab(x)"B[4xspatie]B":printtab(
     x)"B[4xspatie]B":printtab(x)"B[4xspat
     ie]B"
2420 printtab(x)"B[4xspatie]B":printtab(
     x)"JCCCCK":return
2430 rem ** wie heeft gewonnen **
2440 if cpunt<15 and zpunt<15 then return
2450 if zpunt>14 then 2490
2460 pokeb,23:syso:printtab(3)naam$;"
     [RVS-aan]jij hebt gewonnen![RVS-uit]
     [3xspatie]"
2470 pokeb,1:syso:printtab(36)" 15":print
     tab(35)"[neer][RVS-aan]dood"
2480 jij=jij+1:slag$="c":gosub 2960:goto
      2520
```

```
2490 pokeb,23:syso:printtab(10)"[RVS-aan]
     ik heb gewonnen![RVS-uit]"
2500 pokeb,9:syso:printtab(36)" 15":print
     tab(35)"[neer][RVS-aan]dood"
2510 cbm=cbm+1:slag$="s":gosub 2960
2520 print"[CLR-HOME]";tab(13)"totaal-
     score":printtab(13)"EEEEEEEEEEEE"
2530 print"[neer]";tab(3)naam$;tab(27)
     "computer"
2540 printtab(3)"CCCCCCCC";tab(27)"CCC
     CCCCC"
2550 printtab(5)jij;tab(30)cbm
2560 pokeb,23:syso:printtab(3)"tik [RVS-aa
     n]t[RVS-uit](toepen)/[RVS-aan]e[RVS-u
     it](einde)"
2570 get a$:if a$="" then 2570
2580 if a$<>"e" then 2600
2590 print"[CLR-HOME]einde programma":
     end
2600 if cbm+jij>9 then clr:goto 50
2610 print"[CLR-HOME]":cpunt=0:zpunt=0
     :gosub 100:return
2620 rem ** tot.stand/pelt **
2630 print"[HOME]";tab(25)"[neer]comp.
     stand:";str$(cpunt):printtab(26)"
     [neer]slag:";slag
2640 pokeb,9:syso:printtab(25)"splr.st
     and:";str$(zpunt)
2650 printtab(26)"[neer]slag:";slag
2660 if cpunt<>14 and zpunt<>14 then return
2670 if cpunt=14 then pokeb,3:syso:print
     tab(35)"[RVS-aan]pelt"
2680 if zpunt=14 then pokeb,11:syso:print
     tab(35)"[RVS-aan]pelt"
2690 if cpunt=14 and zpunt=14 then return
2700 cklop=2:if cpunt=14 then 1990
2710 if ct<18 and cpunt<11 then a$="n"
     :goto 2330
2720 if ct<18 and cpunt=13 then a$="n"
     :goto 2330
2730 goto 2360
2740 rem ** var.omzetting v.printrouti
     ne **
2750 for p=1 to 4
2760 if comkrt(p)>4 then 2820
2770 if comkrt(p)=1 then ctal$(p)="b"
2780 if comkrt(p)=2 then ctal$(p)="v"
2790 if comkrt(p)=3 then ctal$(p)="h"
2800 if comkrt(p)=4 then ctal$(p)="a"
2810 goto 2840
2820 if comkrt(p)=10 then ctal$(p)="10"
     :goto 2840
2830 ctal$(p)=right$(str$(comkrt(p)),1
     )
2840 next p:return
2850 zpunt=zpunt+cklop:slag$="s":return
2860 cpunt=cpunt+cklop:slag$="c":return
2870 rem ** comp.zoekt kleinste kaart
     **
2880 for p=1 to 4:if comkrt(p)=0 then
     2900
2890 klein=comkrt(p):goto 2910
2900 next p
2910 for p=1 to 4:if comkrt(p)=0 then
     2930
2920 if comkrt(p)<=klein then klein=co
     mkrt(p):c=p
2930 next p:return
2940 for wacht=1 to 1000:next:return
2950 for wacht=1 to 2000:next:return
```

```
2960 for wacht=1 to 7000:next:return
2970 rem ** geluid laag **
2980 for wis=54272 to 54278:poke wis,0
     :next
2990 poke v,15:poke a,45:poke r,60
3000 poke fh,15:poke fl,210
3010 poke g,17
3020 for p=1 to 150:next
3030 poke g,16:poke r,0:return
3040 data 10,"Z",9,"Z",8,"Z",7,"Z",10,
     "X",9,"X",8,"X",7,"X",10,"A",9,"A"
     ,8,"A"
3050 data 7,"A",10,"S",9,"S",8,"S",7,"
     S",4,"Z",3,"Z",2,"Z",1,"Z",4,"X",
     3,"X"
3060 data 2,"X",1,"X",4,"A",3,"A",2,"A"
     ,1,"A",4,"S",3,"S",2,"S",1,"S"

*** EINDE LISTING ***
```

toupen

regel 10	242	regel 560	63
regel 20	44	regel 570	205
regel 30	122	regel 580	67
regel 40	143	regel 590	33
regel 50	165	regel 600	93
regel 60	204	regel 610	230
regel 70	56	regel 620	34
regel 80	109	regel 630	22
regel 90	116	regel 640	218
regel 100	177	regel 650	26
regel 110	75	regel 660	33
regel 120	91	regel 670	25
regel 130	212	regel 680	68
regel 140	183	regel 690	33
regel 150	34	regel 700	179
regel 160	82	regel 710	28
regel 170	174	regel 720	129
regel 180	217	regel 730	46
regel 190	96	regel 740	132
regel 200	154	regel 750	40
regel 210	70	regel 760	210
regel 220	243	regel 770	71
regel 230	25	regel 780	156
regel 240	223	regel 790	39
regel 250	40	regel 800	175
regel 260	61	regel 810	132
regel 270	48	regel 820	51
regel 280	42	regel 830	129
regel 290	31	regel 840	98
regel 300	208	regel 850	239
regel 310	25	regel 860	43
regel 320	151	regel 870	97
regel 330	162	regel 880	247
regel 340	23	regel 890	230
regel 350	99	regel 900	32
regel 360	231	regel 910	76
regel 370	138	regel 920	175
regel 380	242	regel 930	137
regel 390	17	regel 940	104
regel 400	91	regel 950	140
regel 410	132	regel 960	222
regel 420	129	regel 970	66
regel 430	47	regel 980	74
regel 440	110	regel 990	32
regel 450	149	regel 1000	91
regel 460	9	regel 1010	76
regel 470	168	regel 1020	142
regel 480	27	regel 1030	141
regel 490	4	regel 1040	26
regel 500	106	regel 1050	74
regel 510	137	regel 1060	91
regel 520	42	regel 1070	76
regel 530	64	regel 1080	74
regel 540	110	regel 1090	32
regel 550	9	regel 1100	183

reg 1110	39	reg 1870	202	reg 2630	189
reg 1120	181	reg 1880	194	reg 2640	195
reg 1130	186	reg 1890	222	reg 2650	229
reg 1140	40	reg 1900	85	reg 2660	44
reg 1150	146	reg 1910	54	reg 2670	196
reg 1160	16	reg 1920	153	reg 2680	10
reg 1170	250	reg 1930	220	reg 2690	200
reg 1180	230	reg 1940	220	reg 2700	61
reg 1190	203	reg 1950	131	reg 2710	103
reg 1200	77	reg 1960	197	reg 2720	104
reg 1210	5	reg 1970	183	reg 2730	84
reg 1220	7	reg 1980	225	reg 2740	70
reg 1230	62	reg 1990	251	reg 2750	140
reg 1240	82	reg 2000	149	reg 2760	84
reg 1250	212	reg 2010	110	reg 2770	167
reg 1260	15	reg 2020	156	reg 2780	188
reg 1270	63	reg 2030	142	reg 2790	175
reg 1280	63	reg 2040	81	reg 2800	169
reg 1290	72	reg 2050	3	reg 2810	87
reg 1300	210	reg 2060	250	reg 2820	135
reg 1310	108	reg 2070	183	reg 2830	152
reg 1320	25	reg 2080	225	reg 2840	154
reg 1330	12	reg 2090	73	reg 2850	173
reg 1340	40	reg 2100	152	reg 2860	111
reg 1350	51	reg 2110	210	reg 2870	243
reg 1360	129	reg 2120	190	reg 2880	22
reg 1370	19	reg 2130	251	reg 2890	37
reg 1380	186	reg 2140	24	reg 2900	210
reg 1390	36	reg 2150	50	reg 2910	25
reg 1400	115	reg 2160	212	reg 2920	144
reg 1410	34	reg 2170	38	reg 2930	154
reg 1420	63	reg 2180	105	reg 2940	196
reg 1430	32	reg 2190	96	reg 2950	197
reg 1440	32	reg 2200	100	reg 2960	202
reg 1450	110	reg 2210	59	reg 2970	6
reg 1460	160	reg 2220	57	reg 2980	180
reg 1470	101	reg 2230	61	reg 2990	219
reg 1480	50	reg 2240	10	reg 3000	217
reg 1490	216	reg 2250	3	reg 3010	114
reg 1500	79	reg 2260	187	reg 3020	170
reg 1510	144	reg 2270	164	reg 3030	184
reg 1520	237	reg 2280	78	reg 3040	250
reg 1530	142	reg 2290	54	reg 3050	161
reg 1540	219	reg 2300	59	reg 3060	102
reg 1550	28	reg 2310	220		
reg 1560	5	reg 2320	135		
reg 1570	94	reg 2330	163		
reg 1580	113	reg 2340	120		
reg 1590	69	reg 2350	28		
reg 1600	45	reg 2360	202		
reg 1610	162	reg 2370	120		
reg 1620	170	reg 2380	76		
reg 1630	164	reg 2390	155		
reg 1640	92	reg 2400	112		
reg 1650	147	reg 2410	3		
reg 1660	153	reg 2420	41		
reg 1670	62	reg 2430	233		
reg 1680	156	reg 2440	204		
reg 1690	235	reg 2450	184		
reg 1700	143	reg 2460	73		
reg 1710	226	reg 2470	163		
reg 1720	109	reg 2480	41		
reg 1730	183	reg 2490	94		
reg 1740	242	reg 2500	171		
reg 1750	79	reg 2510	151		
reg 1760	57	reg 2520	16		
reg 1770	196	reg 2530	140		
reg 1780	227	reg 2540	192		
reg 1790	202	reg 2550	179		
reg 1800	192	reg 2560	113		
reg 1810	77	reg 2570	155		
reg 1820	156	reg 2580	76		
reg 1830	210	reg 2590	53		
reg 1840	46	reg 2600	57		
reg 1850	255	reg 2610	243		
reg 1860	0	reg 2620	58		

ready.

Morse

Er zijn een vrij groot aantal morse-programma's in omloop. Bij deze software wordt er meestal vanuit gegaan dat iemand het morse alfabet al kent. Nu heb ik wel eens dagen dat ik geen morse gebruik en als dat bij u ook het geval is, kunt u het volgende programma als training gebruiken. Het zet de letter met de bijbehorende streep- en puntcode op het scherm en dit wordt vergezeld door de bekende lange en korte pieptoontjes.

```
  1 rem morse
  2 rem door j van leusen
  3 rem uit alkmaar / 072-121530
  4 rem
 10 print"[CLR-HOME]":c=54272:forl=cto
    c+24:pokel,0:next:pokec+24,15
 20 ready:ify=5then80
 30 n=n+1:poke1024+20*(n-1),n:forv=1to
    y:readx:ifx=0then70
 40 pokev+1027+20*(n-1),45
 50 pokec+5,1:pokec+6,64:pokec+1,142:
    pokec,24:pokec+4,33:fort=0tox:next
 60 pokec+4,32:fort=1to100:next:nextv
    :fort=1to500:next:goto20
 70 pokev+1027+20*(n-1),46:goto50
 80 print"[11xneer]":end
 90 data2,0,99,4,99,0,0,0,4,99,0,99,0
    ,3,99,0,0,1,0,4,0,0,99,99,3,99,99
    ,0,4,0,0,0
100 data0,2,0,0,4,0,99,99,99,3,99,0,9
    9,4,0,99,0,0,2,99,99,2,99,0,3,99,
    99,99,4,0
110 data99,99,0,4,99,99,0,99,3,0,99,0
    ,3,0,0,0,1,99,3,0,0,99,4,0,0,0,99
    ,3,0,99,99
120 data4,99,0,0,99,4,99,0,99,99,4,99
    ,99,0,0,5

*** EINDE LISTING ***
```

```
morse

regel 1    21
regel 2    190
regel 3    155
regel 4    143
regel 10    129
regel 20    244
regel 30    133
regel 40    41
regel 50    37
regel 60    74
regel 70    82
regel 80    82
regel 90    66
regel 100    79
regel 110    102
regel 120    246

ready.
```

Commodore INFO BUSINESS

AMIGA

PC-NIEUWS

DE 68000

ONAFHANKELIJKE INFORMATIE VOOR PROFESSIONELE COMMODORE GEBRUIKERS

Begin maart introduceert Commodore officieel de AMIGA in Europa. In ons land is de persintroductie op 4 maart en op 8 maart kan men in de RAI voor het eerst deze machine op een officiele CBM stand komen bekijken. Die introductie is behoorlijk wat later dan in de Verenigde Staten, en daar is er dan ook al een hele Amiga cultus van de grond gekomen, terwijl we hier nog moeten beginnen. De achtergrond daarvoor is deels gelegen in beschikbare produktiecapaciteit, deels in de PC 10/20, die hier verkocht wordt. Traditioneel had Commodore voor de zakelijke markt in Europa een sterke positie, en die heeft men met de PC 10 aardig uitgebuit. Er was dan ook minder aanleiding om hier snel de Amiga uit te brengen, de softwarehuizen en dealers hadden aan de PC de handen vol. Maar nu is de wonderdoos van Commodore dus ook hier verkrijgbaar en op de HOBBY COMPUTER INFO beurs toont CBM dan ook vol trots zijn paradepaard. We besteden in deze "Commodore Business Info" dan ook veel aandacht aan de Amiga, die we niet als de eindoplossing voor alle kwalen zien, maar als een gerichte probleemoplosser voor beroepen en toepassingen, waarbij de verwerking van gegevens iets meer vraagt dan cijfers en letters op het beeldscherm of papier. De Amiga heeft een grote toekomst als werkstation voor de informatiewerker en als creatieve denkhulp.

Dat men begin maart koos voor de introductie, heeft alles te maken met de CEBIT/Hannover Messe, die van 12 tot 19 maart voor het eerst als separate automatiseringsbeurs plaatsvindt. De Hannover Messe is een plaats, waar Commodore al sinds de dagen van Jack Tramiel zijn grote aankondigingen doet. Deze keer dus de Amiga, die men natuurlijk al een paar dagen eerder aan de pers laat zien, om het "nieuws"-aspect zo veel mogelijk uit te buiten. We verwachten overigens een tamelijk gunstige prijs voor de AMIGA, er kan nu in vergelijking met Amerika en bij de tamelijk gunstige dollarkoers een aantrekkelijk prijskaartje vanaf, denken we.

Het succes van de Amiga hangt, we schreven het al eerder, met name af van de positionering én de MS-DOS compatibiliteit. Het heeft geen zin een dergelijk krachtige, maar ook wat eigenzinnige machine te verkopen als een micro voor iedereen. De gemiddelde gebruiker, die zijn PC gebruikt voor de boekhouding en wat tekstverwerking, heeft helemaal geen behoefte aan al die extra's, aan muisbesturing, afwijkend floppy-formaat en relatief dure kleurenbuizen. Laat hem maar bij z'n MS-DOS machine blijven, die doet voor de algemene toepassingen meer dan voldoende. De Amiga is juist voor hen, die wat meer willen, wat meer met cijfers willen doen, er grafieken van en bij maken, tekst voorzien van illustraties, software voor speciale toepassingen, waarbij geluid en beeld elkaar versterken, voor het "nieuwe" computergebruik.

Dat daarbij enige compatibiliteit met de MS-DOS wereld mooi is meegenomen, spreekt vanzelf. Vooral in kantoren, waar men al PC's heeft, is dat een vereiste, er is tenminste file-compatibiliteit nodig. Gelukkig hebben we nu de eerste MS-DOS pakketten al zien draaien op de Amiga, al wijkt het schijf-format natuurlijk wel af. Maar via de RS-232 verbinding kan men ook al heel wat doen.

Er blijven nog twee zaken, die naar onze mening het succes van de Amiga zouden kunnen versterken. In de eerste plaats goede nederlandse software voor die typisch nederlandse problemen, zoals de boekhouding, het gebruik van de nederlandse lettertekens (dus de vreemde accenten, de ij etc.), de spellingscorrectie en de Viditel-verbindingen. Daarnaast een goede interface naar de wereld van de laserprinters. Programma's als **PageMaker**, waarmee men op een Apple Macintosh werkelijk de grafische fotozetkwaliteit kan benaderen en het begrip **Desktop Publishing** echt inhoud krijgt, moeten er ook voor de Amiga komen. Dan zullen reclamebureau's, drukkerijen, maar ook de huisdrukkerijen van vele bedrijven zich ongetwijfeld op de Amiga storten.

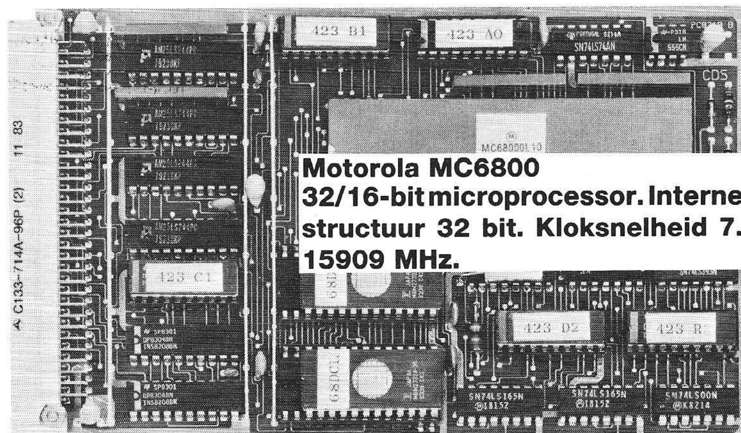
L.Sala ●

Inhoud van dit Business Info gedeelte:

De "oude" CBM 6502-chip en zijn opvolgers vormen het hart van de vertrouwde PET, 8000, Vic-20, C-64, C-16 en de 128. Ondanks zijn goede verdiensten voor menig computeraar bleven er toch nogal wat wensen over. Voor snelle software kon men eigenlijk niet om de machinetaal heen en de adresseerbare geheugencapaciteit blijft beperkt. Daarom koos Commodore voor de Motorola MC68000 microprocessor als het nieuwe brein voor haar innovatieve ST-lijn. Deze 68000-chip is een ware krachtpatser die geheel nieuwe dimensies voor de computergebruikers en -programmeurs opent. In dit artikel een nadere kennismaking met deze veel belovende microprocessor.

De 68000

Een ware krachtpatser.



Motorola MC6800 32/16-bit microprocessor. Interne structuur 32 bit. Kloksnelheid 7. 15909 MHz.

De Motorola MC68000-chip gaat al weer heel wat jaartjes mee. Bij de introductie in 1979 was zij de eerste 16-bit microprocessor die een 32-bit interne architectuur bezat met een 16-bytes niet gesegmenteerde direct memory addressing. Deze hele technische mondvol betekent gewoon dat u in principe toegang heeft tot een totaal van 16. 777. 216 bytes of te wel genoeg informatie om meer dan 5500 getypte A4-tjes te kunnen opslaan. Na de 68000 hebben nog diverse andere krachtige microprocessoren het levenslicht gezien. Geen van hen heeft echter tot nu toe de MC68000 beslissend geklopt. Het 6 bij 7 mm metende en c.a. 70000 transistoren bevattende een siliciumschijfje staat nog immer aan de PC-top en aan de wieg van innovatieve machines als de Amiga en de 520 ST.

8-,16-,32 bits?

De bekende 6502 CPU (is central processing unit) is een rasechte 8-bitter. Dwz kan slechts informatiebrokken van 8 bits (=1byte) op de systeembus zetten of verwerken. Dat schept zo zijn beperkingen, want de onderlinge bus (is systeemverbindings)-lijnen kunnen maar een beperkt aantal "passagiers" vervoeren. Bij een groot BASIC-aanbod treden er dientengevolge vertragingen op. Machinetaal is daarbij de oplossing hetgeen de meeste gebruikers niet erg op prijs stellen.

Computers die grotere bitbrokken aan kunnen bieden een aanzienlijke snelheidswinst bij het gebruik van niet-machinetalen. Deskundige lezers zullen nu opmerken dat de kloksnelheid van de processor eveneens een voorname rol speelt, maar voor de overzichtelijkheid laten wij deze "dienstregeling" even buiten beschouwing.

16-Bitters kunnen twee maal zo veel informatie tegelijkertijd op de systeembus zetten. In theorie zijn dergelijke machines dan ook twee maal sneller als hun 8-bitscollega's. Een 32-bitter kan vier bytes tegelijk aan en is dus vier maal sneller als een 8-bits- en twee maal sneller als een 16-bits processor. Bij de 68000 is, zoals u in het hoofdstukje "geschiedenis" heeft kunnen lezen, nog iets anders aan de hand. De CPU verstuurt weliswaar 2 Byte (=1 woord)-boodschappen per bus, maar doet de "interne post" in 32 bits. Het rekenen in de processor gaat daarom aanzienlijk sneller als de communicatie naar buiten! De vraag of het hier nu om een 16- of 32 bitter gaat laat zich hierdoor niet gemakkelijk beantwoorden. Technisch gezien beschouwen wij de 520 ST echter als een 16 bits-machine.

De 68000 kan maximaal 2 bytes aan informatie per klokcyclus kwijt. Er worden echter ook nog andere datatypen gebruikt. Zelfs vijf verschillende: Een 4-bit binary coded decimal (BCD), 8-bit bytes (B's), 16 bits words (W's) en 32-bit long words (L's). De W's en de L's kunnen uitsluitend op de oneven geheugen adressen opgeslagen worden. Alle overige datatypen zowel op oneven- als evenadressen. De MC68000 beschikt over een groot aantal bevelen om de verschillende datatypen te kunnen manipuleren.

De geschiedenis

De Motorola MC68000-chip gaat al weer heel wat jaartjes mee. Bij de introductie in 1979 was zij de eerste 16-bit microprocessor die een 32-bit interne architectuur bezat met een 16-bytes niet gesegmenteerde direct memory addressing. Deze hele technische mondvol betekent gewoon dat u in principe toegang heeft tot een totaal van 16.777.216 bytes of te wel genoeg informatie om meer dan 5500 getypte A4-tjes te kunnen opslaan.

Na de 68000 hebben nog diverse andere krachtige microprocessoren het levenslicht gezien. Het 6 bij 7 mm metende en c.a. 70000 transistoren bevattende 68000 siliciumschijfje staat nog immer aan de PC-top en aan de wieg van innovatieve machines als de Amiga. Er zijn ook verder ontwikkelingen van uitgekomen, zoals de 68010 en 68020, die wat meer periferie funkties er bij hebben en met name via wat kleine geheugentjes (cache) de adresseringen en de wait-cyclus wat bekorten.

Men heeft de 68000, die niet alleen door Motorola, maar ook door Philips en Hitachi gemaakt wordt, ook in een C-MOS versie gebakken. Dat betekent minder energiegebruik (ongeveer 10% van de normale N-MOS versie) en minder warmteproblemen, dus kan men zo weer snellere versies maken. Bij de nieuwste 68000's zijn de klokfrekwenties al 10 en 12,5 MHz.

Er zijn ook wel concurrenten van de 68000 en vooral IBM is middels haar deelname in Intel natuurlijk een belangrijke factor. In de AT zit bv. de 80286, ook al een vrij krachtige chip, maar Intel heeft met de 80386 al

weer een processor in de 2 á 3 Mips range ter beschikking, waarbij ook nog 4 GigaByte direkt adresseerbaar is en 64 TeraByte indirekt.

De voordelen

De MC6800 is een buitengewoon krachtige microprocessor die de gebruiker en softwaremaker grote voordelen biedt. Een aantal van hen is al terloops de revue gepasseerd. Wij zullen in dit hoofdstukje echter eens wat nader op de vele mogelijkheden ingaan:

❶ De programmeertalen; Het gebruik van de zgn. hogere (dwz meer probleem gerichte) programmeertalen is zonder merkbaar snelheidsverlies mogelijk. C en Pascal vormen geen enkel probleem en dat scheelt de softwareontwikkelaars flink in tijd en geld. Moeizame machinetaalomzettingen behoren bij de 68000 vrijwel tot het verleden.

Heeft BASIC nu afgedaan?

Op het eerste gezicht lijkt het BASIC een verspilling van de 68000-talenten. De hogere programmeertalen halen immers het onderste uit de kan. Toch loont de toepassing van BASIC wel voor de wat eenvoudiger opdrachten of probleembenadering. Bovendien zijn er nieuwe (de zoveelste!) BASIC-versies in ontwikkeling voorzien van extra commando's en methoden om de extra geheugenruimte volledig te kunnen benutten. Verder is de MC 68000 uitstekend geschikt voor het gebruik van GEM- en Logoversies en in een later stadium voor Unix programma's (versie 5.2).

❷ De geheugenruimte; De 6502 chips hebben er tamelijk veel moeite mee om meer dan 64 KRAM vrije geheugenruimte te kunnen benutten. Om de 128 KRAM uit de C128 te krijgen moet de gebruiker op een andere processor mode overschakelen. Ook de Amiga beschikt niet over een ongelimiteerde geheugenruimte. Wel springt deze machine daar zeer efficient mee om. Bovendien heeft de 68000 de potentie om meer dan 16 megabytes te kunnen adresseren! Geheugenvretende programma's zoals HIRES Graphics en spraakherkenning/synthese kan deze CPU danook gemakkelijk aan. Het scheidend vermogen en de graphics van de Amiga slaan alles wat tot nu toe op de PC-markt (tot ongeveer een prijsniveau van 18000 gulden, waar de SUN en Apollo werkstations toch weer meer kunnen) uitgebracht. Zelfs de IBM-PC's kunnen zich eigenlijk nauwelijks met de Amiga meten.

❸ De interne architectuur; Zoals gezegd kan de 68000 CPU de zaken binnenshuis sneller afhandelen als de kontakten met de buitenwereld. Een logische aanpak is daarom de tijd consumerende berekeningen snel intern aftehandelen en ze vervolgens in handzame bytes op de systeembus te zetten. De 68000 beschikt over 56 instructies en 14 address modes. Eigenlijk vrijwel gelijk aan de 6502 chip. Extra zijn de zeventien 32 bitregisters voor algemene doeleinden. Bovendien zijn er ook nog meer dan 1000 enkelvoudige opdrachten. Kortom, de MC68000 beschikt over een krachtig instrumentarium om de aangeboden data snel en op velerlei wijzen te kunnen verwerken.

❹ Twee operating modes; De MC68000 CPU kan in twee verschillende, de gebruikers (=user)- en besturings (=supervisor)-mode werken. Normaliter functioneert het operating systeem in de **supervisor mode** als een soort politieagent. Ernstige gebruikersfouten of mishandeling van het systeem worden tijdig door "vallen" en "ontluizers" onderschept. Illegale instructies of adressering komen zo niet ver. De programma's of talen zelf draaien in de **user mode**. Beide modes maken samen met de hierna besproken interrupts multi tasking mogelijk. Aan de mogelijkheid om meerdere taken tegelijkertijd te kunnen uitvoeren ontleent de Amiga zijn grote kracht.

❺ Interrupts; Interrupts kunnen zoals de naam al aangeeft de uitvoering van een bepaald programma onderdeel of een routine onderbreken. De 68000 benut zgn. vectorinterrupts. Een vector is een geheugenplaats die het adres voor een uitzonderlijke gebruikersroutine bevat. Dat klinkt allemaal veel moeilijker dan het in werkelijkheid is. Het uitzonderlijke (engels exception) betekent niet meer dan dat er tijdens de uitvoering van een programma iets (binnen het programma zelf of van buitenaf) gebeurt. En daarop wordt via een vectorinterrupt gereageerd middels het opentrekken van een geheugenlaatje. De inhoud van dit laatje brengt een bepaalde uitvoeringsroutine ingang, bijvoorbeeld een reset.

Een 68000-machine heeft de beschikking over zeven niveau's van vector-interrupts met elke interrupt zijn eigen adres. M.b.v. deze interrupts kan de 68000 bijzonder snel inspelen op een veranderende situatie of een gebruikersfout ondervangen. De 68000-microprocessor beschikt over drie interruptlijnen. Alle drie zijn zij betrokken bij een interruptverzoek. Daarmee is het mogelijk om de interrupts in zeven belangrijkheidstrappen op te delen. Het ene interruptverzoek krijgt dus meer prioriteit dan het andere hetgeen een afhandeling van zaken in volgorde van belangrijkheid waarborgt.

Over de mogelijkheden en voordelen van de Motorola MC68000 raak je eigenlijk niet uitgepraat. Helaas ontbreekt ons de ruimte om meer dan een indruk te kunnen geven. Maar waarom ontdekt u hen het niet zelf op de binnenkort leverbare magnifieke Commodore Amiga? ●

Technische gegevens

De 14 Addressing Modes:
1. Dataregister direct
2. Adresregister direct
3. Register indirect
4. Register indirect met post-increment
5. Register indirect met pre-decrement
6. Register indirect met displacement
7. Register indirect met index
8. Absolute short
9. Absolute long
10. PC relative met displacement
11. PC relative met index 12. Immediate
13. Quick Immediate
14. Implied register

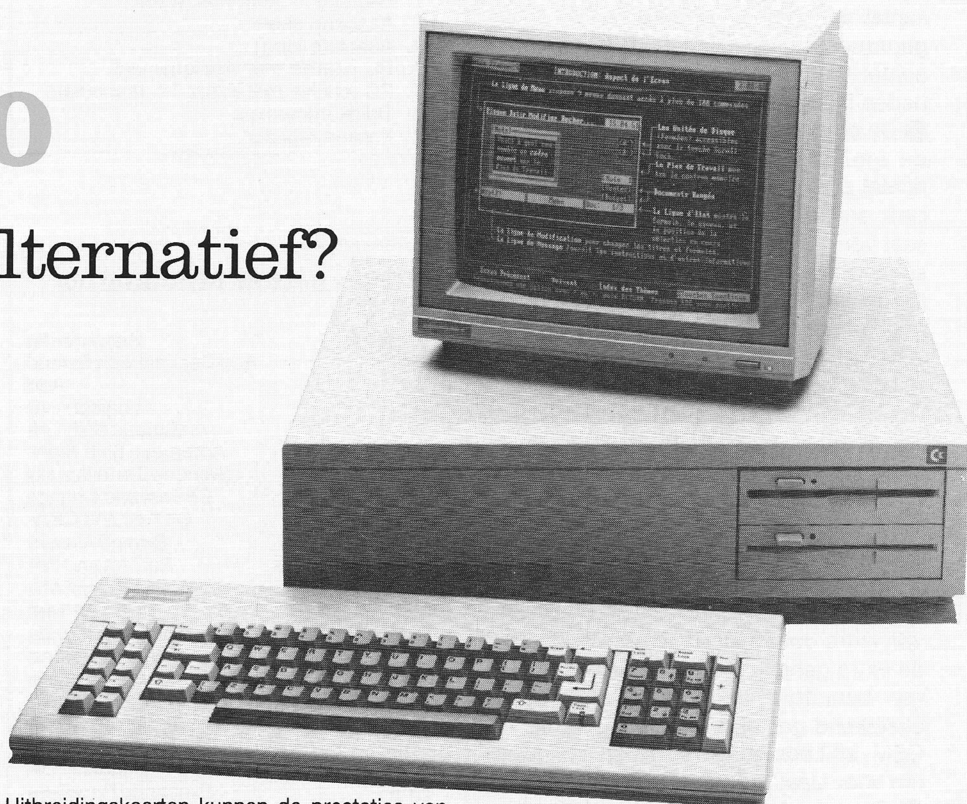
Instruction Mnemonics

Mnemonic	Beschrijving
ABCD	Add Decimal with Extend
ADD	Add
AND	Logical AND
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
BCC	Branche Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMP	Compare
DBCC	Test Cond,Decrement and Branch
DIVS	Signed Divide
DIVU	Unsigned Devide
EOR	Exclusive OR
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral Data
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical OR
PEA	Push Effective Address
RESET	Rest External Devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
ROXR	Rotate Right with Extend
RTE	Return from Exception
RTR	Return from Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
SCC	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap on Overflow
TST	Test
UNLK	Unlink

Door alle drukte rondom de Amiga en de C-128 zijn de Commodore PC-10 en -20 wat op de achtergrond geraakt. En dat is jammer want deze IBM-compatibele PC's hebben tegen een aantrekkelijke prijs toch behoorlijk veel in hun mars. Of de aanschaf van een PC-10 voor de meer serieus georiënteerde hobbyist loont, is een kwestie van afwegen. De PC krijgt in ieder geval in Commodore Business Info alle aandacht.

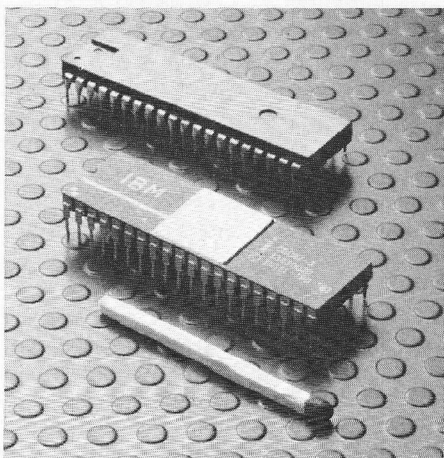
De PC-10

Het serieuze alternatief?



Uitbreidingskaarten kunnen de prestaties van de PC opvijzelen en aanpassen voor vrijwel elke toepassing.

Commodore is geen onbekende op de markt voor zakelijke 16- bitters. Na geruime tijd een eigenzinnige koers met de CBM 3000-, 4000 en 8000-serie gevaren te hebben, kwamen ruim een jaar geleden de PC-10 en diens grotere broer de PC-20 op de Nederlandse markt. Bij de introductie was de PC-10, met een toenmalig prijskaartje van rond de *f* 6000,- (inclusief BTW), al een geduchte IBM- concurrent. Met name in het kleinbedrijf was de half zo dure PC-10 een aantrekkelijk alternatief voor de IBM-machines. De prijs is ondertussen nog fors lager, al ging ook IBM wat omlaag. Wel zijn er vele alternatieven bijgekomen in de prijsklasse tussen de 3 en 4000 gulden.
Inmiddels is namelijk het voorbeeld van Commodore door vele andere leveranciers van de zogenaamde IBM-klonen gevolgd. Er zijn zelfs fabrikanten die voor iets meer dan tweeduizend gulden een complete IBM-compatibele PC presenteren.



De 8080 en de optionele 8087 co-processor, die het hart van de PC vormen.

Toch is het wel degelijk oppassen geblazen met dergelijke klonen. De in advertenties met veel poeha geroemde compatibiliteit blijkt nogal eens beduidend minder dan 99,9% of men heeft diverse moge lijkheden van de oorspronkelijke PC-uitvoering gestript. En daar kom je als niets vermoedende koper vaak te laat achter!
In de praktijk blijken de meeste IBM-klonen betrouwbare machines. De electronica laat niet meer verstek gaan dan bij de IBM- fabrikaten. Alleen de mechanische uitvoering kan wel eens wat minder degelijk uitpakken, de koeling is soms wat mager, zeker als men ook op de voeding beknibbeld heeft.
Ook de Commodore PC-10 is een goede IBM-compatibele PC en de recente prijsdalingen bieden de serieuze computeraar een veelzijdige machine voor minder dan *f* 4000,-.
Voor een C-128 met twee drives, een geheugenuitbreiding tot 256 KB en een monochrome monitor betaalt men al gauw hetzelfde. Een volledig opgetuigde Amiga kost bijna het dubbele, verwachten we.
Of een PC-10 het werkpaard voor u is, hangt van de gewenste toepassing(en) af. En welke voordelen biedt zo'n Commodore PC nu eigenlijk?

De uitvoering

De PC-10 ziet er in de standaarduitvoering net zo uit als de doorsnee (IBM-)Personal Computer. Een systeemkast met twee 360K drives, een los toetsenbord met spiraalkabel en een op de kast te plaatsen groen-monitor. Gewoon een doelmatige machinebouw.
De metalen systeemkast meet 39x49x15cm. Een behoorlijk bureau is daarbij geen overbodige luxe. Overigens nemen een C-128 (33x43cm) en twee diskdrives naast elkaar meer plaats in. De meeste gebruikers zullen die drives echter onder het bureaublad plaatsen zodat de systeemkast toch meer werkbladruimte inneemt.
Een systeemkast biedt de gebruiker meer gemak dan hij of zij zich op het eerste

gezicht realiseert. Zo zitten de beide 360K drives handzaam boven elkaar aan de rechter kastzijde. Bukken en andere lastige houdingen bij het schijfwisselen behoren daarmee tot het verleden. Uitbreiding van de mogelijkheden vergt geen com plete demontage en solderen of gehannes met cartridges meer. Want in zo'n systeemkast kan gemakkelijk een uitbreidingskaart in een van de connectorslots geplaatst worden.
Verder oogt een systeemkast professioneler als een vol met apparatuur gestouwd computermeubel en is de kabelspaghetti keurig weggewerkt.
Het toetsenbord is praktischer ingedeeld dan bij de IBM-PC's. De plaatsen van de SHIFT- en RETURN-toetsen zijn veranderd en het numerieke deel kreeg een aparte ENTER-toets. Ook extra zijn de indicatielampjes voor de CAPS- en NUMERIC-LOCK-toetsen. Als enig nadeel geldt wellicht, dat het Commodore bord minder solide aanvoelt.
De 12-inch groenmonitor van Philips is goed leesbaar, hoewel het contrast bij daglicht wat tegenvalt. Een monitor, die tot de middenklasse behoort.

De drives

De PC-10 is standaard uitgerust met een dubbele 360K diskdrive. Bij de C-128 kan de gebruiker kiezen uit de enkelzijdige 1570 (168K)- en de dubbelzijdige 1571 (336K) drives. Op het eerste gezicht geen grote verschillen. De dataoverdracht bij de PC-10 gaat echter aanzienlijk sneller dan bij de C-128 en C-64. Ondanks het feit dat de C-128 onder CP/M kan draaien, blijft diens snelheid magertjes.
Vergeleken met de nieuwste PC's is de diskopslagcapaciteit van 2 maal 360 KB niet indrukwekkend. 700K- of 1 Megabyte drives vormen geen uitzondering meer. De kleingebruiker en hobbyist hebben in de praktijk voor de meeste applicaties aan de geboden 2 x 360K ruim voldoende.

De mogelijkheden

De grote kracht van de PC-10 ligt op het terrein van de serieuze toepassingen. Bijvoorbeeld tekstverwerking, calculatie, grafische ontwerpen, boekhouding, databeheer en communicatie. De Intel 8088 microprocessor met een kloksnelheid van 4,77 MHz maakt de PC-10 tot een veelzijdige machine, die meer dan 99% van de IBM-PC software vlekkeloos draait. Letterlijk alles wat de serieuze gebruiker begeert, is onder MS-DOS te krijgen.

De PC-10 beschikt standaard over 256K RAM-geheugen. In vergelijking met een aantal nieuwere PC-machines een geringe RAM- capaciteit maar wel twee maal zo groot als het C-128-RAM. En voor ca. ƒ 500,- is een uitbreiding tot 640K intern geheugen te koop.
De systeemkast bezit vijf slots voor uitbreidingskaarten. De monochrome monitorkaart neemt er daarvan een in beslag. Zij die kleur willen, zullen apart een kleurenkaart en -monitor moeten aanschaffen. In tegenstelling tot de IBM-machines gaan deze slots niet op aan allerlei besturingskaarten. Uitbreidingsmogelijkheden als schijf-, video- en printerbesturing zijn namelijk al op de hoofdkaart aangebracht. De gebruiker staan dus minimaal drie vrije slots ter beschikking om de PC-10 volgens de eigen specificaties op te tuigen.
De IBM-PC's staan nu niet bepaald als snelheidsmonsters te boek. Dat zelfde geldt idem dito voor de PC-10. Voor kleinen hobbygebruik is deze Commodore Personal Computer echter snel genoeg. Zeker in vergelijking met de C-128- en C-64 huiscomputers. Wie meer snelheid wil praat wel over machines die al gauw driemaal zo duur als de PC-10 zijn. Eventueel is inbouw van een 8087-coprocessor voor het snelle rekenwerk mogelijk.
Wat de in- en output betreft, beschikt de PC-10 over vier aansluitingen: Parallel Centronics, RS232 serieel, toetsenborden videoaansluiting. Lastige problemen bij het aansluiten van printers of modems doen zich daarbij zelden voor. De C-64-bezitters zijn dat wel anders gewend.
Op de systeemhardware heeft Commodore jammer genoeg toch wat beknibbeld. Er is bijvoorbeeld geen ingebouwde klok/kalender. Ook bezit de PC-10 geen BASIC-ROM, ook al om copyright problemen te omzeilen. Het PC-10-BASIC dient de gebruiker zelf in te laden. Het 8K Rom bevat slechts de interrupts- en opstartroutines plus het MS-DOS 2.0-2.11. Of u behalve BASIC ook andere programmatuur bij de PC-10 krijgt, hangt van de leverancier af. **Soms houdt men de DOS buiten de prijs, dat scheelt weer wat in de lokprijs!**
Een belangrijk punt; de graphics en het geluid. Grafisch staat de PC-10, mits met goede monitor, zijn mannetje met een hoge schermresolutie en vele kleurmogelijkheden. Qua geluid doet deze machine echter duidelijk onder voor de C-128/64 en de Amiga. Maar wilt u muziek op kantoor?

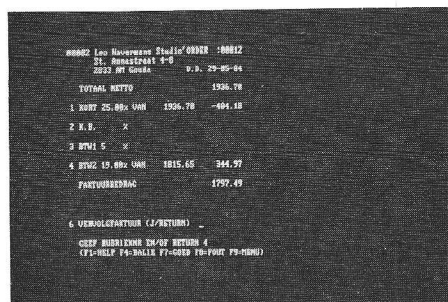
De software

De grote kracht van een IBM-PC-kloon schuilt hem in de hoeveelheid leverbare zakelijke- en industriële software. Vroeger was de IBM-PC-programmatuur stevig aan de prijs en viel daarom buiten het bereik van het hobbybudget. De laatste tijd is ook de PC-software aan een sterke prijsdruk onderhevig en liggen professionele programma's voor minder dan helft van de originele prijs in het schap.
Een prijs-/kwaliteitsvergelijking tussen PC- en huiscomputer- programmatuur is geen eenvoudige zaak. Al gauw worden er appels met peren vergeleken. Een professionele tekstverwerker biedt bijvoorbeeld meer mogelijkheden en bedieningsgemak dan een amateurversie. Of de incidentele brievenschrijver nu bij zo'n professioneel pakket gebaat is, moet betwijfeld worden. De liefhebbers van videospelletjes komen bij een IBM-PC aanzienlijk minder aan hun trekken.
In Amerika worden goed lopende games wel in een IBM-PC- versie uitgebracht. De Nederlandse import laat echter danig ver-



De PC/AT met de 80286 processor is weer sneller dan de PC. We wachten op een Commodore versie van deze machine

stek gaan. Een betreurenswaardig feit gezien de toch wel hoge schermresolutie van de PC-10. Verder zullen de zelf programmeurs van spelletjes natuurlijk de sprites missen.
De liefhebbers van grafische ontwerpen kunnen hun hart bij de PC- 10 ophalen. De kleurenbeelden met hoge schermresolutie



Nederlandstalige software is zeker voor financiële pakketten aan te bevelen. Overzichtelijke schermen, zoals dit van IBM's "Facturering" met b.t.w. en dergelijke helpen de "leek"/gebruiker.

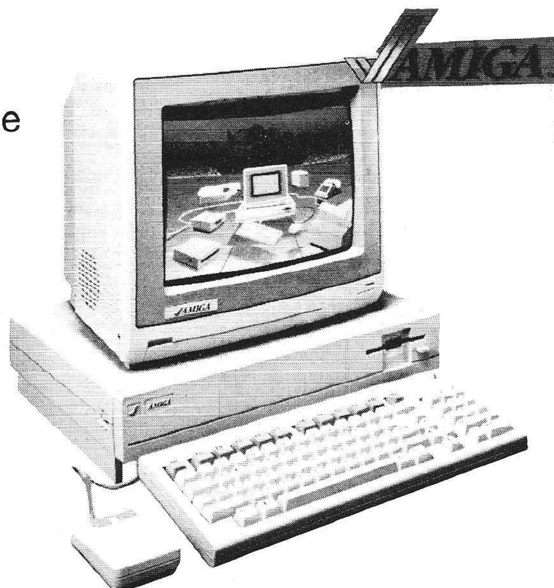
kunnen zich met de Amiga (640x352 versus 640x400) meten, maar de snelheid is dan wel minder. Ook zijn er diverse CAD- en tekenpakketten leverbaar. Alleen bij de animaties moet de PC-10 zijn meerdere in de Amiga erkennen.

Nabeschouwing

De PC-10 is een serieuze, zij het goedkope computer en dus geen spelletjesmachine. Liefhebbers van videogames zijn aanzienlijk veelzijdiger en voordeliger uit met een C-64. Op het gebied van tekstverwerking en zakelijke toepassingen kan de 40 koloms C-64 niet goed meer meekomen. Ook de nieuwe C-128 is in wezen een conservatieve huiscomputer. Het CP/M-besturingssysteem werkt t.g.v. de gesloten compromissen te traag om de verwende zakelijke gebruiker te kunnen bevredigen. En het prijsverschil tussen een volledig opgetuigde C-128 en een PC10 valt eigenlijk te verwaarlozen. Bovendien laat de verkrijgbaarheid van zakelijke CP/M-programmatuur voor de C-128 momenteel (nog) te wensen over.
De aantrekkelijkheid van de C-128 schuilt in de mogelijkheid om zo wel alle C-64-software als CP/M zakelijke programmatuur te kunnen draaien en het krachtige C-128-BASIC. Gebruikers die laag geprijsde softwarepakketten en -spelletjes zoeken, graag zelf in BASIC programmeren en/of af en toe een zakelijk programma willen runnen, kunnen zich zeker in de C-128 vinden. Een ander belangrijk argument voor de aanschaf van de C-128 is, dat de gebruiker het systeem langzaam kan opbouwen vanuit de C-64 spullen. Velen zullen niet zo maar het hele bedrag voor een PC-10 op tafel kunnen leggen.
De PC-10 is een machine, die de zakelijke- of serieuze computeraar zal aanspreken. Voor een redelijk bedrag krijg je een machine, die zich qua mogelijkheden en snelheid duidelijk onderscheidt van een C-128 of C-64. Zij die de spelletjes node missen, kunnen als aanvulling misschien een C-64 met datarecorder in de uitverkoop op de kop tikken.

Liefhebbers van graphics, animatie en geluid zijn beter uit met een Amiga. Helaas zal diens huidige prijsstelling menig hobbybudget overschrijden. Gaat het alleen om gedetailleerde grafische platen, dan is een PC-10 met extra kleurkaart voldoende.
De PC-10 doet het nog steeds goed in de professionele sector. Of zo'n PC'tje ook iets voor u is, hangt van het gebruik af. Wellicht is het feit, dat in de VS de rol van de 8-bitter eigenlijk uitgespeeld is, tekenend voor de toekomst van de 16-bits machines, zoals de PC-10 en de Amiga. De tijd zal leren of de door de hobbyist wat vergeten PC ook in Nederland koning huiscomputer van diens troonzal stoten.

De Rijksuniversiteit Leiden was de gastheer bij de persintroductie op 4 maart van de AMIGA. Interesse vanuit de universitaire wereld voor de grafische kwaliteiten van deze machine vormde daarvoor de aanleiding.



Vriendelijke Amiga

Het eerste verkoopoffensief van Commodore richt zich op de technisch-wetenschappelijke wereld. Daar moeten de kwaliteiten als werkstation de onderzoekers en rekenmeesters gaan omturnen tot grootgebruikers van de Amiga. Gerichte marketing is dus het devies van Commodore en op 4 maart had men dan ook erg veel universitaire grootheden uitgenodigd. Het gebeuren vond plaats in het universitaire rekencentrum van de R.U. Leiden aan de Wassenaarse weg, de juiste plaats voor grensverleggende informatica.

Maar wat ons natuurlijk het meest interesseert, is hoever de grenzen van de Amiga op dit moment dan wel liggen. De basisbeschrijving van deze machine hebben we reeds vele malen in dit blad geplaatst. Daar willen we u niet nog een keer mee vermoeien, of anders kunt u die informatie ook elders nog wel lezen. De superieure graphics, spraak- en geluidsfunkties van de Amiga zijn tenslotte pas van belang, als er software beschikbaar is, die daar gebruik van maakt. Misschien zijn er wat hobbyisten, die zich in de verborgen rijkdommen van het Metacomco operating Systeem van de Amiga willen verdiepen, voor de gewone gebruiker is dat nauwelijks van belang. Wie Mandelbrot graphics op de Amiga wil zetten, kan daarmee zeer fijne plaatjes toveren, maar 99% van de gebruikers kent dit soort wiskundige beelditeratie-grapjes niet eens.

MS-DOS draaien

De gewone gebruiker wil weten, of hij z'n PC software kan draaien op de Amiga. Wel, dat kan via de MS-DOS emulator van Smile Research inderdaad. Maar wel met enige hindernissen, en de potentiële koper van de Amiga moet daar goed op letten. De eerste horde was de vertaling van de verschillende MS-DOS commando's naar Amiga commando's. Die horde is (wat betreft normale tekst/data software zonder bijzondere beeldgrapjes) goeddeels genomen met de genoemde emulator, maar dat wil nog niet zeggen dat alle MS-DOS software zomaar draait. Sommige programma's gaan namelijk ietsje buiten de MS-DOS specificatie om, wanneer ze direkt de processor aanspreken om bijvoorbeeld een snelle sorteerroutine of beeldvorming te verkijgen. Die dingen vinden dan in de Amiga niet de instruktieset van de 8088, maar van de 68000 en dat gaat dus niet goed.
Een tweede obstakel is de schijfformattering. De 3.5 inch schijfjes van de Amiga zijn zelfs voor de leek verschillend van de standaard 5 1/4 inch diskettes van de PC wereld. IBM heeft wel plannen om ook 3,5 duims drives te gebruiken, en sommige concurrenten zoals Philips met de YES doen dat al, maar software op 3.5 inch is zeldzaam. Nu is dat probleem op te lossen door aan de Amiga een of meerdere 5¼ inch drives te hangen en dan is, met de emulator, software wel in te lezen. Voor wie MS-DOS compatibiliteit dus hoog in het vaandel voert, is de aanschaf van extra 5 1/4 inch drives dus te overwegen.
Niet dat we verwachten, dat alle software dan probleemloos draait, want er is ook nog zoiets als kopieerbescherming. Die kan de Amiga in MS-DOS mode nog danig parten spelen.

De Amiga kan dus MS-DOS software aan, maar interne conversie naar Amiga-Dos is (nog) niet mogelijk.

Het is dus niet mogelijk, om bv. Lotus files, die in de MS-DOS mode zijn ingelezen, over te halen naar de Amiga mode, daar dan mee te werken en de zaak weer terug te brengen naar MS-DOS formaat. Op dit moment is door de emulator de Amiga a.h.w gesplitst in twee aparte machines, die niet met elkaar communiceren. Maar er wordt gewerkt aan een brugfunktie voor dit probleem.

Parallel communicatie

Uitwisselen van data tussen de PC en de Amiga kan ook via datacommunicatie en de RS 232. Veel sneller werkt nog de koppeling via parallele poorten. Daarmee zijn effectief 10 KByte/s over te sturen en de software daarvoor is via "Public Domain" uit de VS al beschikbaar. We hebben overigens veel geleerd over de mogelijkheden om met de Amiga te werken uit vele megaBytes aan informatie in de diverse databanken en mailboxen, die we konden inzien. Daaruit blijkt, dat vooral de wetenschappers de Amiga als zeer interessant werktuig zien.

Lotus

Ondertussen zijn er natuurlijk wel een aantal MS-DOS pakketten, die draaiend op de Amiga zijn gedemonstreerd, en Lotus 1-2-3 is daarvan de belangrijkste. Dit superieure spreadsheet-pakket is voor veel PC gebruikers dagelijkse kost en wanneer dat op de Amiga draait, is men vaak al meer dan tevreden. Maar we vinden wel, dat de emulatie behoorlijk wat snelheid kost en het beeld op de monitor ook vrij slecht overkomt, niet de fijne IBM matrix in ieder geval. Ook **Wordperfect** hebben we al op de Amiga tot leven gebracht gezien. Overigens houdt deze compatibiliteit ook in, dat de gegevens van veel meer software-pakketten overdraagbaar zijn. Zelfs al draait een bepaald pakket niet op de Amiga, men kan dan de gegevens of files in ieder geval wel bewerken met de tekstverwerker etc. Pakketten als **Textcraft** van Commodore zijn als tekstverweker namelijk helemaal gericht op het benutten van de Amiga mogelijkheden. Dat uitwisselen van files

(binnen de MS-DOS mode) is dan vrij simpel, tenminste wanneer het gaat om files, die een ASCII, DIF, dBase of ander standaard codering gebruiken.

Fun
Commodore zelf wil op dit moment niet de nadruk leggen op het recreatieve gebruik van de Amiga. Spelletjes zijn er wel, maar men praat liever over serieuze toepassingen. In ieder geval wil men een spelletjesimago vermijden. Maar in de creatieve sfeer zijn doe-programma's natuurlijk ook erg leuk. Software juweeltjes als **De Luxe Paint** van Electronic Arts uit San Mateo in Californië en **Graphicraft** en **Musicraft** van Commodore zelf zijn misschien erg op hun plaats in studio's, we vinden ze voornamelijk erg leuk. Zelf beeldkunst maken, met geluiden en spraak experimenteren, dat is amusement. En hoewel de prijzen van die pakketten wel boven die voor de 64/128 liggen, is een niveau van 2 tot 300 gulden per pakket niet overdreven.

Programmeertalen
De professionele computergebruiker kan met de Amiga al behoorlijk uit de voeten, er zijn al heel wat talen beschikbaar. **Pascal, Modula-2, Lisp**, om er maar een paar te noemen. Maar er is ook al een **C-compiler** en wel de 3.03 versie van Lattice. daarmee is porting van C-code naar b.v. de PC al erg gemakkelijk geworden, een enorm voordeel voor ontwikkelaars.

Laserprinters
We komen nog even terug op onze wens, om direkt een laserprinter aan te kunnen sturen met de Amiga. Nu kan dat via de parallel of RS-232 poort vrijwel zonder problemen. Bijvoorbeeld de HP Laserjet is daarvoor te gebruiken. Het probleem zit 'm niet in de overdracht van normale teksten die via de ISO-ANSI terminal codes overgedragen kunnen worden. Ook zijn via de instelbare "Preference Code" printer drivers al wel beperkte screendumps te maken, zo zijn grijswaarden te vertalen in hardcopy, hetgeen bij tekenpakketten wel handig werkt. Echte software pakketten voor "desktop publishing" moeten er nog komen.

●

Infocom tekst-adventures

Het Amerikaanse softwarehuis Infocom (bekend van oa. Zork) heeft een aantal van haar bekenste tekst-adventures voor de Amiga bewerkt.
Het gaat voornamelijk om interactieve Science Fiction verhalen zoals bijvoorbeeld het nu al befaamde **The Hitchikers Guide To The Galaxy**. In dit humoristische toekomstverhaal lift de laatste overlevende van de per ongeluk vernietigde aarde, Arthur Dent, door het heelal en belandt daarbij in diverse krankzinnige situaties. Om zijn overlevingskansen te vergroten is Arthur Dent gewapend met de liftersgids, een gevaar opsporende zonnebril, een "geen-paniek-knop", een microscopische ruimtevloot, een pakje Miltipurpose Fluff en vernietigingsorders.
Andere Infocom Science Fiction Amiga-adventures zijn o.a. **Planetfall, Starcross, Suspended** en **A Mind Forever Voyaging.**
Blijft de vraag waarom Infocom tekstadventures uitbrengt in plaats van de sublieme Amiga graphics te benutten? Sterontwerper Brian Moriarty antwoordt daarop met een tegenvraag: "Waarom zijn niet alle boeken geillustreerd?" Antwoord: De lezer maakt liever eigen gedachtenhelden en datzelfde geldt ook voor de tekstadventures. Bovendien blijft bij het achterwege laten van de graphics meer ruimte over om het verhaal uit te diepen.

Tecmar randapparatuur

Nu de eerste Amiga's al bij de trotse eigenaren staan te snorren, begint de stroom aan compatibele randapparatuur op gang te komen. Tecmar, een bekende Amerikaanse leverancier van PC- peripherals, biedt de Amigagebruiker de volgende producten:
- Het **multifunctie uitbreidingsmodule T-Card,** dat het Amigageheugen tot 1MB vergroot. Verder zitten er op de T-Card een kalenderklok met standby batterijvoeding, een seriële-poort, een parallel- of SASI-poort, een gebufferde busexpansie-poort en een eigen netvoeding.
- **De 20MB Hard Disk, T-Disk.** Deze krachtige 3.5inch diskdrive wordt op de Amiga zelf geplaatst, waardoor zij geen bureauruimte in beslag neemt. Een speciale afgeschermde kabel verbindt de harde schijf met de SASI-poort van de T-Card. Verder zijn er indicatie-LED's voor power, select en write.
- **De 20MB Tape Backup, 20MB T-Tape.** Deze snelle datadrive maakt in enkele minuten een copie van de 20MB van T-Disk en is aanzienlijk goedkoper dan een extra 20MB T-Disk. Er zijn indicatie-LED's voor het tracknumber, tapedirection, read, write, door en power. T-Tape gebruikt speciale enkelspoels tapecartridges van een zeer goede kwaliteit.
- **Het 2400 Baud Modem, T-Modem** met keuze uit de overdrachtsnelheden 300-, 1200- en 2400 Baud. Of dit modem ook voor het Nederlandse telefoonnet aangepast zal worden, is nog niet bekend.

De Tecmar randapparatuur is als een bouwdoos systeem ontworpen. De T-Card komt vertikaal naast de rechterzijde van de Amiga te staan. De T-Disk, T-Tape en T-Modem staan bovenop elkaar naast de monitor en gedeeltelijk op de Amigasysteemkast en T-Card. Voor verdere info Tecmar, 6225 Cochran Road, Solon, Ohio 44139, USA.

Aegis videoanimatie en painting

Aegis Development Inc, Santa Monica, heeft onlangs een videoanimatie- en een paintprogramma voor de Amiga uitgebracht:
Aegis Animator creëert bewegende videobeelden in full colour. Een intern graphics systeem stelt de operateur in staat zelf beelden op te bouwen en over het gehele monitorscherm te manipuleren. Draaien, links/rechts, op/neer en in/uit zijn daarbij mogelijk. De aldus gemaakte graphicsbeelden worden vervolgens op een storyboard geplaatst waarna het editen met behulp van cut-, splice- en pastfuncties kan beginnen. Tot slot kunnen de beelden van bijpassend geluid voorzien en op de videorecorder opgenomen worden.
Aegis Images is een paintpakket voor het scheppen van computerkunst of het maken van achtergronden voor de Aegis Animator. De animator kan zowel met 320 x200- als 640-200 graphicsschermen werken tot een maximum van 32 (met een keuze uit 4096) kleuren tegelijk. De ware artiest kan zich helemaal uitleven met wel 20 verschillende electronische penselen, een airbrush met verstelbaar spuitstuk, de opties spray, glow, wash, smear, gradient fill, en dithering ("bevende lijnen"), enkel- en meervoudige color cycling enz.
De gemaakte graphicsbeelden kunnen verder nog vergroot, geroteerd, bijgesneden en gekopieerd worden. Verder beschikt Aegis Images over standaardvormen/figuren als lijnen, curven, veelhoeken, cirkels, punten, parallelogrammen, bogen, driehoeken, noem maar op. De geboden mogelijkheden zijn schier onbeperkt.
De Animator en Images zijn producten van Aegis Development Inc, 2210 Wilshire Blvd, Suite 277, Santa Monica, CA 90403, USA.

Mini-Orgel

Er zit meer muziek in uw Commodore dan u denkt, maar misschien zit er in u ook wel meer muziek dan u denkt. Met dit computerorgeltje kunt u zich gaan verbazen over zowel de Commodore-64 als uzelf. Na het intikken heeft u in ieder geval altijd prijs; er komt altijd geluid uit...

```
  1 rem mini-orgel
  2 rem door o. vos
  3 rem uit amersfoort / 033-806550
  4 rem
 10 print"[CLR-HOME][1.rood]":poke532
    80,0:poke53281,0:s=54272:d=s+7:a=
    0
 20 print"[3xspatie]O‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾
    ‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾P"
 30 print"[3xspatie]|  [GEEL][2xspatie]
    electronisch[2xspatie]mini orgel[3xsp
    atie][1.rood]  |"
 40 print"[3xspatie]L________________
    ________________√"
 50 print"[2xneer]"
 60 print"[7xrechts]O‾PO‾PO‾PO‾PO‾PO‾
    PO‾PO‾PO‾P"
 70 print"[7xrechts]|  ||  ||  ||  ||  ||
     ||  ||  ||  |"
 80 print"[7xrechts]L_√L_√L_√L_√L_√L_
    √L_√L_√L_√"
 90 print"[8xrechts]|  ||  ||  ||  ||  ||
     ||  ||  | "
100 print"[8xrechts]|  ||  ||  ||  ||  ||
     ||  ||  | "
110 print"[8xrechts]L_√L_√L_√L_√L_√L_
     L_√L_√ "
120 print"[HOME][8xneer][grijs 3][8xrecht
    s]a[2xrechts]s[2xrechts]d[2xrechts]
    f[2xrechts]g[2xrechts]h[2xrechts]
    j[2xrechts]k[2xrechts]l"
130 print"[2xneer][9xrechts]z[2xrechts]
    x[2xrechts]c[2xrechts]v[2xrechts]
    b[2xrechts]n[2xrechts]m[2xrechts]
    ,"
140 print"[2xneer][15xrechts]=wavevor
    m"
150 print"[neer][3xrechts]+ = tweeste
    mmig[3xspatie][RVS-aan][1.blauw]-
    [RVS-uit][grijs 3] = eenstem"
160 pokes+4,64:pokes+6,240:pokes+4,16
    :ifa=1then180
170 goto500
180 poked+4,64:poked+6,240:poked+4,16
190 pokes+24,0
200 geta$:ifa$=""then200
210 ifa$="a"thenfr=2000:goto440
220 ifa$="s"thenfr=2225:goto440
230 ifa$="d"thenfr=2450:goto440
240 ifa$="f"thenfr=2675:goto440
250 ifa$="g"thenfr=2900:goto440
260 ifa$="h"thenfr=3125:goto440
270 ifa$="j"thenfr=3350:goto440
280 ifa$="k"thenfr=3575:goto440
290 ifa$="l"thenfr=3800:goto440
300 ifa$="z"thenfr=4025:goto440
310 ifa$="x"thenfr=4250:goto440
320 ifa$="c"thenfr=4475:goto440
330 ifa$="v"thenfr=4700:goto440
340 ifa$="b"thenfr=4925:goto440
350 ifa$="n"thenfr=5150:goto440
360 ifa$="m"thenfr=5375:goto440
370 ifa$=","thenfr=5600:goto440
380 ifa$="1"then500
390 ifa$="2"then510
400 ifa$="3"then520
410 ifa$="-"thena=0:print"[HOME][16xneer]
    [3xrechts][grijs 3]+[17xrechts][RVS-a
    an][1.blauw]-[RVS-uit][grijs 3]":
    goto180
420 ifa$="4"then530
430 ifa$="+"thena=1:print"[HOME][16xneer]
    [3xrechts][RVS-aan][1.blauw]+[RVS-uit
    ][grijs 3][17xrechts]-":goto160
440 rem---------set up sid-----------
450 pokes+24,15:pokes+6,240:hf=fr/256
    :lf=(fr+1000)/256
460 ifa=1thenpoked+6,240:poked+4,33
470 pokes+1,hf:pokes,(hf-int(hf))*256
480 ifa=1thenpoked+1,lf:poked,(lf-int
    (lf))*256
490 forw=0to150:next:goto190
500 pokes+4,19:print"[HOME][14xneer][1.bl
    auw][3xrechts][RVS-aan]1[RVS-uit]
    [grijs 3][rechts]2[rechts]3[rechts]
    4":goto190
510 pokes+4,33:print"[HOME][14xneer][grij
    s 3][3xrechts]1[rechts][1.blauw][RVS-
    aan]2[RVS-uit][grijs 3][rechts]3[rech
    ts]4":goto190
520 pokes+4,97:print"[HOME][14xneer][grij
    s 3][3xrechts]1[rechts]2[rechts][RVS-
    aan][1.blauw]3[RVS-uit][grijs 3][rech
    ts]4":goto190
530 pokes+4,129:print"[HOME][14xneer]
    [grijs 3][3xrechts]1[rechts]2[rechts]
    3[rechts][RVS-aan][1.blauw]4[RVS-uit]
    [grijs 3]":goto190

*** EINDE LISTING ***
```

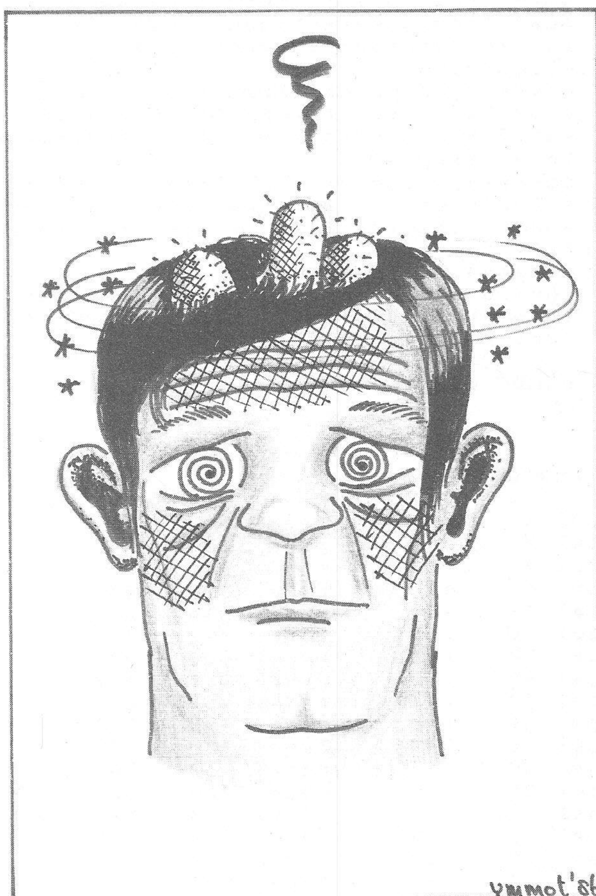
mini-orgel

regel 1	98
regel 2	56
regel 3	173
regel 4	143
regel 10	105
regel 20	165
regel 30	152
regel 40	148
regel 50	255
regel 60	174
regel 70	102
regel 80	133
regel 90	55
regel 100	55
regel 110	109
regel 120	89
regel 130	238
regel 140	102
regel 150	76
regel 160	226
regel 170	30
regel 180	140
regel 190	86
regel 200	95
regel 210	53
regel 220	80
regel 230	65
regel 240	76
regel 250	68
regel 260	69
regel 270	71
regel 280	81
regel 290	73
regel 300	87
regel 310	85
regel 320	73
regel 330	83
regel 340	72
regel 350	75
regel 360	83
regel 370	41
regel 380	83
regel 390	85
regel 400	87
regel 410	131
regel 420	89
regel 430	229
regel 440	132
regel 450	202
regel 460	88
regel 470	54
regel 480	122
regel 490	13
regel 500	36
regel 510	187
regel 520	197
regel 530	241

ready.

Kop van Jut

Laat u niet door de titel van het programma afschrikken, er hoeft namelijk niet met een extreem grote hamer op uw computer te worden geslagen. Het is de bedoeling om na het startsein met uw joystick, die in poort twee moet worden aangesloten, kloksgewijs te gaan draaien. De draaisnelheid en tijdsduur die u gebruikte zal uiteindelijk bepalend zijn voor de kracht waarmee het gewicht omhoog gestuwd wordt bepalen. U heeft drie pogingen om de **kop van jut** te raken. Mocht u hierin slagen dan krijgt u opnieuw drie beurten maar wordt het niveau moeilijker. Mocht de **kop van jut** u bekend voorkomen dan moet u eens naar de gedigitaliseerde foto van onze hoofdredacteur kijken in **Luc Sala's datakolom.**




```
  1 rem kop van jut
  2 rem door marc de hingh
  3 rem uit breda / 076-145652
  4 rem
 10 sys65409:v=53248:pokev+32,15:poke
    v+33,15:poke646,11:printchr$(8)chr$
    (14)
 20 print"'Kop van Jut'":print" (Kop
    van de uitgever)":print"[neer]Eve
    n Geduld..."
 30 fori=15872to16127:readx:pokei,x:next
    :fori=16128to16255:pokei,0:next
 40 poke16130,15:poke16133,3:poke1619
    2,255:poke16193,192:poke16195,254
 50 fori=16256to16319:pokei,0:next:for
    i=0to7:poke16256+i*3,255:next
 60 fori=16384to16495:readx:pokei,x:next
 70 dims$(11):fori=0to11:reads$:s$(i)
    =chr$(128+asc(s$))+mid$(s$,2)+"!"
    :next
 80 fori=0to5:pokev+39+i,0:next:pokev
    +45,15:pokev+46,3:pokev+14,172:poke
    v+15,226
 90 fori=0to6:poke2040+i,248+i:next:poke
    2047,254:pokev,149:pokev+2,173
100 pokev+4,149:pokev+6,173:pokev+8,1
    49:pokev+10,173:pokev+1,55:pokev+
    3,55
110 pokev+5,76:pokev+7,76:pokev+9,97:
    pokev+11,97:pokev+13,82:pokev+21,
    255
120 print"[HOME][5xneer][bruin]":fori
    =0to16:printtab(18)"[RVS-aan][2xspati
    e]":next
130 printtab(16)"[RVS-aan][oranje][6xspat
    ie]":printtab(16)"[RVS-aan][6xspatie]
    [RVS-uit]";
140 pokev+12,0:print"[HOME][4xneer][grijs
     1]Vuurknop=Start":gosub370:print
    "[op][14xspatie][neer]"
150 print"[BLAUW]poging nr.[2xspatie]
    123":print"score:[3xspatie]000000"
160 print"hoogste: 000000[op]":pg=1:s
    c=0:m=250:gosub340
170 print"[HOME][4xneer][BLAUW][RVS-aan]
    [4xspatie]T I J D[4xspatie]":poke
    v+12,144:poke55546+pg,6:poke55547
    +pg,0
180 s$="Actie!!!":gosub310:sys16384:s
    =peek(251)+256*peek(252):sc=sc+10
    *s
190 gosub340
200 s=s/m:h=226:b=1:hh=h
210 pokev+15,h:h=h-s:s=s-.05:ifh<99then
    h=99:s=-s-1:sys16448:b=0
220 ifh<hhthenhh=h
230 ifh<226then210
240 pokev+15,226
250 onbgoto280:s$="Aaauw!":gosub310:for
    t=1to1000:next:sc=sc+10000:gosub3
    40
260 s$="Nu ":ifm>250thens$=s$+"nog "
270 s$=s$+"moeilijker ":gosub310:m=m+
    rnd(1)*100:goto290
280 s$=s$((hh-99)/11):gosub310
290 pg=pg+b:fort=1to1000:next:ifpg<4goto
    170
300 s$="G A M E[2xspatie]O V E R":gosub
    310:goto140
310 print"[RVS-uit][ZWART][HOME][5xneer]"
    spc(21);:fori=1tolen(s$):forj=1to
    70:next:printmid$(s$,i,1);
320 next:fort=1to300:next
330 fori=1tolen(s$):forj=1to50:next:print
    chr$(20);:next:return
340 sc$=mid$(str$(sc),2):print"[HOME]
    [7xneer][BLAUW]"tab(15-len(sc$))s
    c$
350 ifsc>hithenhi=sc
360 hi$=mid$(str$(hi),2):printtab(15-
    len(hi$))hi$:return
370 wait56320,16:wait56320,16,16:return
380 data0,0,1,0,0,3,0,0,13,0,0,63,0,0
    ,127,0,0,238,0,1,240,0,1,229,0,3,
    235,0
390 data7,207,0,15,143,0,15,7,0,15,1,
    0,15,0,0,15,0,0,15,0,0,15,0,0,15,
    0,0,7
400 data0,0,7,0,0,3,31,0,166,0,0,255,
    192,0,255,248,0,255,252,0,255,254
    ,0,127
410 data255,0,191,255,128,255,255,192
    ,255,255,224,255,255,224,223,255,
    224,255
```

```
420 data255,240,235,255,240,4,31,240,
    0,15,240,0,7,240,0,7,240,0,11,240
    ,0,29
430 data240,0,63,240,0,14,240,0,0,9,1
    91,0,4,159,0,5,183,0,1,131,0,0,14
    3,0,2
440 data64,0,2,32,0,2,30,0,2,0,0,1,0,
    0,1,0,0,0,0,0,0,0,0,3,0,0,4,0,0
    ,0,0,0
450 data0,0,0,64,0,0,96,0,0,48,0,0,25
    ,0,134,251,224,171,253,224,199,12
    7,192
460 data79,185,192,79,255,192,15,101,
    128,15,131,128,15,199,128,15,255,
    128,135
470 data255,0,63,255,0,31,255,0,7,254
    ,0,255,254,0,253,252,0,15,252,0,2
    55,248
480 data0,255,248,0,15,240,0,39,240,0
    ,183,224,0,0
490 data169,0,133,251,133,252,169,14,
    133,253,169,0,133,254,169,0,133,1
    62,173
500 data0,220,41,31,197,2,240,6,230,2
    51,208,2,230,252,133,2,165,162,20
    1,4,48
510 data233,206,12,208,230,254,165,25
    4,201,8,208,218,166,253,169,32,15
    7,160
520 data4,198,253,16,203,96
530 data120,162,81,160,64,142,20,3,14
    0,21,3,169,96,133,251,88,96,198,2
    51,208
540 data10,162,49,160,234,142,20,3,14
    0,21,3,162,10,189,0,208,73,1,157,
    0,208
550 data202,202,16,244,76,49,234
560 datahahahaha,wat jammer nou,helaas,
    je komt er wel,ai ai,je kunt het niet
570 dataslecht hoor,pech gehad,je lee
    rt 't nooit,heel zwak,actie zei ik
580 datawakker worden

*** EINDE LISTING ***
```

```
 kop van jut

 regel 1     81
 regel 2     221
 regel 3     15
 regel 4     143
 regel 10    30
 regel 20    45
 regel 30    44
 regel 40    41
 regel 50    70
 regel 60    162
 regel 70    33
 regel 80    44
 regel 90    169
 regel 100   97
 regel 110   66
 regel 120   229
 regel 130   204
 regel 140   202
 regel 150   17
 regel 160   117
 regel 170   126
 regel 180   154
 regel 190   36
 regel 200   67
 regel 210   86
 regel 220   71
 regel 230   90
 regel 240   195
 regel 250   44
 regel 260   3
 regel 270   220
 regel 280   89
 regel 290   67
 regel 300   118
 regel 310   11
 regel 320   103
 regel 330   149
 regel 340   181
 regel 350   227
 regel 360   124
 regel 370   223
 regel 380   241
 regel 390   254
 regel 400   106
 regel 410   232
 regel 420   52
 regel 430   27
 regel 440   228
 regel 450   67
 regel 460   240
 regel 470   133
 regel 480   75
 regel 490   157
 regel 500   119
 regel 510   115
 regel 520   58
 regel 530   165
 regel 540   73
 regel 550   39
 regel 560   129
 regel 570   243
 regel 580   23
 ready.
```

Competitie

Een competitie hoeft niet direkt het eerste elftal van een voetbalclub te betreffen. U kunt ook thuis een competitie houden, met deelnemers en standen. In wezen een kleine database, en het volgende programma laat u standen en uitslagen met behulp van uw computer bijhouden.

```
  1 rem competitie
  2 rem door j. bijlsma
  3 rem 2e exloermond / 05997-1475
  4 rem
 10 printchr$(147):poke 53280,0:poke
    53281,0
 20 gosub 1300: rem *** korte uitleg
    ***
 30 printchr$(5);"-- moet er een best
    and ingelezen worden[4xspatie]van
    af tape (j/n)"
 40 get a$:if a$<>"j" and a$<>"n" then
     40
 50 if a$="j" then u$="j":goto 1080:
    rem ** inlezen **
 60 u$="n":printchr$(147)
 70 input"[ROOD]-- hoeveel ploegen in
     competitie ";a
 80 dim pl$(a),sp(a*2),gw(a*2),gl(a*2
    ),vl(a*2),pu(a*4):printchr$(147)
 90 for x= 1 to a
100 print chr$(28);"-- geef naam van
    ploeg ";x
110 printchr$(5):input pl$(x)
120 if len(pl$(x))>11 then print:print
    "-- invoer te lang (max. 11 lette
    rs)":goto 110
130 printchr$(147):next
140 printchr$(28);"[9xspatie]*** over
    zicht ***[2xspatie]";chr$(5)
150 for t= 1 to 2000:next :printchr$(
    147)
160 print"ploeg";tab(12)"aw";tab(17)"
    gew";tab(22)"gel";tab(27)"ver";tab(
    32)"pun"
170 print
180 for x=1 to a
190 print pl$(x);tab(12)sp(x);tab(17)
    gw(x);tab(22)gl(x);tab(27)vl(x);tab(
    32)pu(x)
200 print chr$(28)"-------------------
    ---------------------";chr$(5)
210 q=q+2:if q>17 then gosub 360
220 next
230 gosub 360
240 printchr$(147)
250 if u$="n" then print"zijn punten,
    stand enz al ingevoerd (j/n)"
260 if u$="j" then 290
270 get a$:if a$<>"j" and a$<>"n" then
     270
280 if a$="n" then gosub 410: rem ***
    ** eenmalige opstart-invoer *****
290 print"-- wilt u uitslagen invoere
    n (j/n)":print
300 get a$:if a$<>"j" and a$<>"n" then
     300
310 if a$="j" then printchr$(147):gosub
     540: rem *** uitslagen invoeren
    ***
320 print"-- wilt u de nieuwste resul
    taten zien[6xspatie](j/n)"
330 get a$:if a$<>"j" and a$<>"n" then
     330
340 if a$="j" then 670: rem *** uitsl
    agen invoeren ***
```

```
350 print:print:goto 140
360 rem ******* toets routine *******
370 print:print:q=0
380 print chr$(28);"[8xspatie]*** dru
    k op een toets ***"
390 get a$:if a$="" then 390
400 printchr$(5):return
410 rem **** invoeren van gegevens **
    **
420 printchr$(147)
430 for x=1 to a
440 printchr$(28);"-- ploeg[3xspatie]
    : ";pl$(x):print:print
450 input"[WIT]-- hoeveel gespeeld ";
    sp(x)
460 if sp(x)=0 then printchr$(147):return
470 input"-- hoeveel gewonnen ";gw(x)
480 input"-- hoeveel gelijk[3xspatie]"
    ;gl(x)
490 input"-- hoeveel verloren ";vl(x)
500 if gw(x)+gl(x)+vl(x)<>sp(x) then
    print:print"-- foutieve invoer":print
    :goto 440
510 pu(x)=gw(x)*2+gl(x)
520 printchr$(147):next
530 printchr$(147):u$="j":return
540 rem ***** uitslagen invoeren ****
    *
550 for x= 1 to a
560 sp(x)=sp(x)+1
570 print"[ROOD]";pl$(x);" :[WIT]";"[RVS-
    aan]w[RVS-uit]inst, [RVS-aan]v[RVS-ui
    t]erlies of [RVS-aan]g[RVS-uit]el
    ijk"
580 print:print"-- voer 0 in als ";pl
    $(x):print"[3xspatie]niet heeft g
    espeeld"
590 get a$:if a$<>"g" and a$<>"v" and
     a$<>"w" and a$<>"0"then 590
600 if a$="0" then printchr$(147):sp(
    x)=sp(x)-1:next
610 if a$="w" then gw(x)=gw(x)+1
620 if a$="v" then vl(x)=vl(x)+1
630 if a$="g" then gl(x)=gl(x)+1
640 print chr$(147)
650 next
660 printchr$(147):return
670 rem ********   afdrukken ********
680 printchr$(147)
690 printchr$(156);
700 print"ploeg";tab(12)"aw";tab(17)"
    gew";tab(22)"gel";tab(27)"ver";tab(
    32)"pun"
710 printchr$(5):print
720 for x=1 to a
730 pu(x)=gw(x)*2+gl(x)
740 print pl$(x);tab(12)sp(x);tab(17)
    gw(x);tab(22)gl(x);tab(27)vl(x);tab(
    32)pu(x)
750 print chr$(28)"-----------------
    -------------------";chr$(5)
760 q=q+2:if q>17 then gosub 360
770 next
780 gosub 360
790 printchr$(147)
800 print"-- opslaan op tape (j/n)
810 geta$: if a$<>"j" and a$<>"n" then
    810
820 if a$="n" then 290
830 print:print:print"-- gegevens voo
    r tape-opslag"
840 print:input"-- datum ";d$
850 print:input"-- bestandsnaam ";b$
860 print:print"-- bestand ";b$;" wor
    dt weggeschreven"
870 rem **** wegschrijven naar tape *
    **
880 open 1,1,1,b$
890 print#1,d$;chr$(13)
900 print#1,a ;chr$(13)
910 for x=1 to a
920 print#1,pl$(x);chr$(13)
930 print#1,sp(x) ;chr$(13)
940 print#1,gw(x) ;chr$(13)
950 print#1,gl(x) ;chr$(13)
960 print#1,vl(x) ;chr$(13)
970 print#1,pu(x) ;chr$(13)
980 if st <> 64  then next
990 close 1
1000 print chr$(147)
1010 print"-- bestand ";b$;" weggeschr
     even":print
1020 print"-- bestand bevat ";a;" gege
     vens"
1030 gosub 360
1040 printchr$(147):print"-- programma
      beeindigen (j/n)"
1050 geta$:ifa$<>"j" and a$<>"n" then
     1050
1060 if a$="j" then end
1070 goto 140
1080 rem ***** inlezen van tape ******
1090 printchr$(147)
1100 input"-- welk bestand inlezen (na
     am) ";b$
1110 open 1,1,0,b$
1120 input#1,d$
1130 input#1,a
1140 for x=1 to a
1150 input#1,pl$(x)
1160 input#1,sp(x)
1170 input#1,gw(x)
1180 input#1,gl(x)
1190 input#1,vl(x)
1200 input#1,pu(x)
1210 if st <> 64  then next
1220 close 1
1230 printchr$(147)
1240 print"-- bestand ";b$;" zit in he
     t geheugen":print
1250 print"-- bestand bevat ";a;" gege
     vens":print
1260 print"-- bestand is bijgewerkt to
     t ";d$
1270 gosub 360
1280 goto 140
1290 rem ****** korte uitleg *******
1300 print"met dit programma kunt u ee
     n willekeuri-ge competitie bijhou
     den"
1310 print"nadat u de eerste keer de s
     tand,punten[2xspatie]enz hebt ing
     evoerd ";
1320 print"hoeft u de daaropvol-gende
     weken slechts in te voeren of de
     [2xspatie]betrokken ";
1330 print"ploegen hebben gewonnen,ver
     lo-ren of gelijlkgespeeld."
1340 print"de computer werkt dan alle
     andere gege- vens automatisch af"
1350 gosub 360
1360 printchr$(147):return
```

*** EINDE LISTING ***

competitie

regel 1	130	regel 30	130
regel 2	61	regel 40	55
regel 3	62	regel 50	162
regel 4	143	regel 60	68
regel 10	172	regel 70	87
regel 20	101	regel 80	136

```
regel 90    161
regel 100    23
regel 110    14
regel 120    153
regel 130    9
regel 140    220
regel 150    97
regel 160    251
regel 170    153
regel 180    161
regel 190    138
regel 200    149
regel 210    44
regel 220    130
regel 230    38
regel 240    77
regel 250    188
regel 260    134
regel 270    108
regel 280    30
regel 290    201
regel 300    102
regel 310    91
regel 320    30
regel 330    105
regel 340    75
regel 350    196
regel 360    144
regel 370    217
regel 380    210
regel 390    105
regel 400    174
regel 410    126
regel 420    77
regel 430    161
regel 440    239
regel 450    0
regel 460    117
regel 470    14
regel 480    88
regel 490    30
regel 500    118
regel 510    11
regel 520    9
regel 530    8
regel 540    69
regel 550    161
regel 560    37
regel 570    236
regel 580    141
regel 590    117
regel 600    38
regel 610    255
regel 620    6
regel 630    217
regel 640    77
regel 650    130
regel 660    21
regel 670    202
regel 680    77
regel 690    136
regel 700    251
regel 710    185
regel 720    161
regel 730    11
regel 740    138
regel 750    149
regel 760    44
regel 770    130
regel 780    38
regel 790    77
regel 800    4
regel 810    108
regel 820    118
regel 830    148
regel 840    20
regel 850    8
regel 860    136
regel 870    144
regel 880    28
regel 890    20
regel 900    237
regel 910    161
regel 920    21
regel 930    248
regel 940    243
regel 950    232
regel 960    247
regel 970    250
regel 980    41
regel 990    209
regel 1000    77
regel 1010    248
regel 1020    249
regel 1030    38
regel 1040    70
regel 1050    153
regel 1060    87
regel 1070    30
regel 1080    129
regel 1090    77
regel 1100    123
regel 1110    27
regel 1120    73
regel 1130    34
regel 1140    161
regel 1150    74
regel 1160    45
regel 1170    40
regel 1180    29
regel 1190    44
regel 1200    47
regel 1210    41
regel 1220    209
regel 1230    77
regel 1240    226
regel 1250    204
regel 1260    92
regel 1270    38
regel 1280    30
regel 1290    0
regel 1300    250
regel 1310    9
regel 1320    246
regel 1330    192
regel 1340    76
regel 1350    38
regel 1360    21

ready.
```

Codewoord

Als u langzamerhand een beetje uitgekeken bent op het spel Mastermind, dan hebben we hier een nieuwe uitdaging voor u. Het spel Codewoord is lijkt nogal op Mastermind maar in plaats van cijfercombinaties werkt het met woorden. Dat klinkt niet al te moeilijk, met de klemtoon op klinkt.

```
 10 rem codewoord / cbm 64
 20 rem door r.a. wolswijk
 30 rem zeist / 03404-23295
 40 rem
 50 dim cw$(100):gosub 490
 60 gosub 380:a$="":input"[2xneer]cod
    ewoord d.m.v. de computer (j/n)";
    a$
 70 if a$<>"j" then 110
 80 z=int(100*rnd(0)+1):if cw$(z)=""
    then 80
 90 w$=cw$(z):cw$(z)="":w=w+1:if w>90
     then w=0:restore:gosub 490:goto
    80
100 goto 150
110 gosub 380:w$="":input"[2xneer]cod
    ewoord (4 letters)";w$
120 if len(w$)=4 then 150
130 print"[RVS-aan][neer]niet meer of
     minder dan 4 letters s.v.p!"
140 for wacht=1 to 4000:next:goto 110
150 rt=1110:xl=1091:jl=1098:gosub 380
160 t=1:for pr=1 to 10:p=0:x1=0:x2=0:
    xl=xl+80:jl=jl+80:rt=rt+80
170 if pr=10 then t=0
180 print:print:printtab(t);str$(pr);
    ".[10xspatie]R R R R[8xlinks]";
190 p=p+1:if p>4 then gosub 470
200 get rw$(p):if rw$(p)="" then 200
210 if p>4 and rw$(p)=chr$(13) then 2
    60
220 if p>1 and rw$(p)=chr$(20) then print
    "[links]R[2xlinks]";:p=p-1:gosub
    460:goto 200
230 if p>4 then 200
240 if asc(rw$(p))<65 or asc(rw$(p))>
    90 then 200
250 print"[rechts]";rw$(p);:goto 190
260 rem ** letters op juiste plaats *
    *
270 l=0:gosub 450
280 for k=1 to 4:if lw$(k)=rw$(k) then
     l=l+1:rw$(k)="":lw$(k)=" ":gosub
     410
290 next k:if l=4 then print"[neer]":
    goto350
300 rem ** alleen juiste letters **
310 for kk=1 to 4:for k=1 to 4
320 if rw$(k)=lw$(kk) then rw$(k)="":
    lw$(kk)=" ":gosub 430
330 next k,kk,pr:gosub 450:print
340 print"[neer][RVS-aan]codewoord wa
    s:";:for p=1 to 4:print lw$(p);" "
    ;:next:print".";
350 a$="":input"[RVS-aan]opnieuw (j/n
    )";a$
360 if a$<>"n" then 60
370 print"[CLR-HOME]einde programma":
    end
380 print"[CLR-HOME][RVS-aan]--------
    ---------------------------------"
390 print"[RVS-aan][10xspatie]***code
    woord***[2xspatie]VVVV[3xspatie]+
    +++ "
400 print"[RVS-aan]------------------
    ------------------------[op]":return
410 rem ** pinnetjes letters op juist
    e plaats **
420 poke xl+x1,81:poke xl+x1+40,66:x1
    =x1+1:return
430 rem ** pinnetjes juiste letters *
    *
440 poke jl+x2,81:poke jl+x2+40,66:x2
    =x2+1:return
450 for sp=1 to 4:lw$(sp)=mid$(w$,sp,
    1):next
460 for wis=rt to rt+5:poke wis,32:next
    :return
```

```
470 poke rt,146:poke rt+1,133:poke rt
    +2,148:poke rt+3,149:poke rt+4,14
    6
480 poke rt+5,142:return
490 rem ** codewoorden**
500 for in=1 to 100:read cw$(in):next
    :return
510 data "veel","meel","kaas","nimf",
    "balk","zeem","kast","baas","kaal"
    ,"keel"
520 data "boek","boom","kust","zaak",
    "kies","inge","toen","koen","most"
    ,"mond"
530 data "lont","boeg","boog","mast",
    "hugo","eelt","dijk","rijk","bijl"
    ,"koel"
540 data "vast","muur","kuur","zuur",
    "buur","huif","zeug","bram","koos"
    ,"boos"
550 data "wijn","zalf","bank","poot",
    "land","ziek","bier","shag","bies"
    ,"lies"
560 data "hier","beek","voet","zalm",
    "lamp","been","hout","koud","snel"
    ,"loof"
570 data "disk","flop","vlag","erwt",
    "stom","gids","puls","eind","show"
    ,"wesp"
580 data "vera","ooit","ruit","kiem",
    "riem","vlas","wolf","mars","trap"
    ,"elsa"
590 data "henk","staf","dorp","vork",
    "kolk","voer","melk","klos","jong"
    ,"tram"
600 data "weer","geel","rood","kras",
    "moer","bout","spar","aard","harp"
    ,"blad"
*** EINDE LISTING ***
```

```
codewoord

regel 10    160
regel 20    44
regel 30    122
regel 40    143
regel 50    138
regel 60    169
regel 70    27
regel 80    153
regel 90    15
regel 100   31
regel 110   252
regel 120   61
regel 130   77
regel 140   84
regel 150   44
regel 160   237
regel 170   29
regel 180   105
regel 190   246
regel 200   113
regel 210   74
regel 220   140
regel 230   249
regel 240   182
regel 250   59
regel 260   146
regel 270   142
regel 280   30
regel 290   180
regel 300   223
regel 310   147
regel 320   150
regel 330   144
regel 340   253
regel 350   79
regel 360   243
regel 370   53
regel 380   138
regel 390   149
regel 400   119
regel 410   66
regel 420   242
regel 430   222
regel 440   218
regel 450   4
regel 460   206
regel 470   111
regel 480   167
regel 490   112
regel 500   27
regel 510   31
regel 520   150
regel 530   104
regel 540   17
regel 550   97
regel 560   129
regel 570   205
regel 580   180
regel 590   179
regel 600   101

ready.
```

INGS - C-16 LISTINGS - C-16 LISTINGS - C-16 LISTINGS

Morse

Mocht u een cursus morse willen volgen dan hoeft u niets meer te doen dan dit negentien regels tellende programma in te tikken. En wat avonden tijd reserveren natuurlijk.

```
 1 rem morse / c16 en plus4
 2 rem door durk bouma
 3 rem uit nijland / 05156-550
 4 rem
 5 rem ******************************
   *****
 6 forf=1to8:keyf,"sys62116"+chr$(13
   ):next:trap7:dimm(36):vol8:print"
   [CLR-HOME]"
 7 resumenext:data22222,12222,11222,
   11122,11112,11111,21111,22111,222
   11,22221
 8 data12,2111,2121,211,1,1121,221,1
   111,11,1222,212,1211,22,21,222,12
   21,2212
 9 data121,111,2,112,1112,122,2112,2
   122,2211
10 forf=1to36:readm(f):next:color1,7
   ,4
11 input"[neer]tekst";z$:print:forf=
   1tolen(z$):o$=mid$(z$,f,1):n=asc(
   o$)
12 ifn<48orn>90then19
13 ifn>57andn<65then19
14 m=n-47:ifm>10thenm=m-7
15 print" "o$" ";:m$=str$(m(m)):forg
   =2tolen(m$):n$=mid$(m$,g,1)
16 ifn$="1"thensound1,850,5:color4,3
   :color0,3:print".";:goto18
17 sound1,850,10:color0,6:color4,6:print
   "-";
18 forp=1toval(n$)*125:next:next
19 print:forp=1to125:nextp,f:goto11

*** EINDE LISTING ***
```

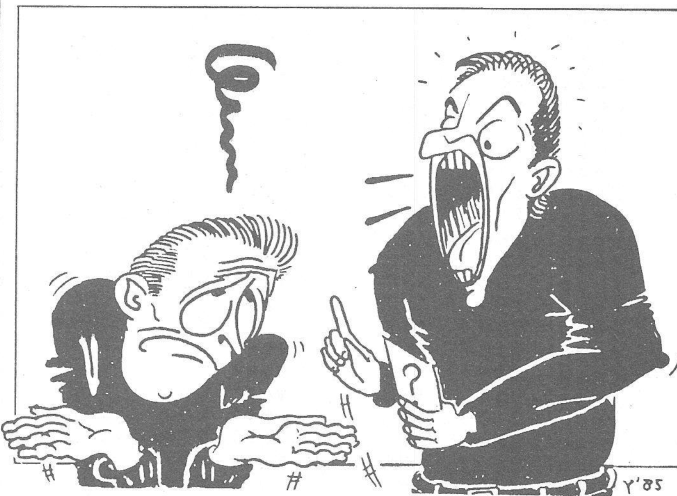
```
morse

regel 1     21
regel 2     190
regel 3     155
regel 4     244
regel 10    129
regel 20    244
regel 30    133
regel 40    41
regel 50    37
regel 60    74
regel 70    82
regel 80    82
regel 90    66
regel 100   79
regel 110   102
regel 120   246

ready.
```

Masterbrain

Een computervariant van het spel Mastermind. Het werkt via de bekende spelregels, maar ondanks dat blijft het toch een boeiend spel om te spelen. De algorithme is zo gekozen, dat de C- 16 (of Plus/4) een combinatie van getallen in z'n geheugen neemt, die door de speler moet worden geraden. Erg aardig is het daarbij, dat je geen dubbele getallen kunt invoeren. Soms was vier keer hetzelfde getal intikken voldoende om te weten te komen, of dat bewuste getal in de geheime combinatie voorkomt. De C-16 steekt daar in dit programma een stokje voor. Een leuk stuk programmatuur, waarin ook de geluidschip van de C-16 niet is vergeten.



```
 10 rem masterbrain com. c-16 en plus
     4
 20 rem d. la brijn tel. 070-541033
 30 rem printen spelregels
 40 printchr$(27)+chr$(77):color4,2,5
    :color0,8,7
 50 a$="[ROOD][RVS-aan][40xspatie]"
 60 b$="[RVS-aan][8xspatie][RVS-uit][23xs
    patie][RVS-aan][9xspatie]"
 70 c$="[RVS-aan][8xspatie][RVS-uit]
    m a s t e r b r a i n [RVS-aan][9xspa
    tie]"
 80 d$="[RVS-aan][23xspatie]"
 90 print"[CLR-HOME]";a$;b$;c$;b$;a$
100 print"[neer][rechts]probeer 4 ver
    schillende getallen te"
110 print"[rechts]raden tussen 0 en 1
    0 en zet ze in de":print"[rechts]
    goede volgorde."
120 print"[neer][rechts]type 4 versch
    illende getallen in":print"[rechts]
    tussen 0 en 10."
130 print"[neer][rechts]je krijgt 15
    beurten."
140 print"[neer][rechts]druk toets 1,
     2 of 3 in voor de"
150 print"[rechts]achtergrondkleur en
     voor de start."
160 print"[neer][rechts]1=geel[5xspatie]
    2=groen[5xspatie]3=grijs[neer]"
170 printa$;a$;a$;a$;a$
180 rem keuze achtergrondkleur
190 getkeyx$:g=int(val(x$)):ifg<1org>
    3then190
200 on g goto 210,220,230
210 h=8:i=7:j=10:o=1:p=6:q=1:goto250
220 h=13:i=0:j=4:o=7:p=8:q=7:goto250
230 h=2:i=6:j=7:o=2:p=6:q=3:goto250
240 rem starten spel + printen kopreg
    el
250 dima(4),b(4):vol7:t=16:v=0:w=0
260 color0,h,i:color4,h,i:color1,j,o
270 print"[CLR-HOME][rechts]getypte",
    "[2xrechts]aantal","goede","[3xrechts
    ]beurt"
280 print"[rechts]getallen","[2xrechts]
    goed","plaats","[3xrechts]nummer"

290 print"[rechts][ROOD]▬▬▬▬▬▬▬▬▬▬▬▬▬
    ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
300 c=0:f=0:x=0
310 rem genereren 4 verschillende get
    allen
320 forl=1to4:a(l)=int(1+9*rnd(0)):next
    l
330 form=1to3:forn=m+1to4
340 ifa(m)=a(n)then320
350 nextn:nextm
360 do until f=4
370 c=c+1:rem aantal beurten
380 ifc=16thenv=v+1:x=6:goto760:rem n
    iet geraden
390 ifc=15thenx=4:gosub760:rem laatst
    e beurt
400 rem input getallen + controle
410 b(1)=10:b(2)=11:b(3)=12:b(4)=13
420 fork=1to4
430 getkeye$:b(k)=val(e$)
440 ifx>0thenprint"[CLR-HOME][2xHOME]"

450 x=0
460 forl=1to3:form=l+1to4
470 ifb(l)=b(m)thenx=1
480 nextm:nextl
490 ifb(k)<1thenx=2
500 ife$="0"thenx=3
510 ifx>0thengosub740
520 ifx>0thengoto430
530 rem wisselen kleuren + uitprinten
     input
540 s=p:ifc/2=int(c/2)thens=j
550 ifc/5=int(c/5)thens=q
560 color1,s,o:print"[HOME][2xneer]"
570 forn=1toc:printspc(40):nextn
580 printtab(2*k-2)b(k):sound1,800,3
590 nextk
600 forl=1to300:next:sound1,380,10
610 rem beoordeling input
620 e=0:f=0
630 forl=1to4:form=1to4
640 ifa(l)=b(m)thene=e+1
650 nextm
660 ifa(l)=b(l)thenf=f+1
670 nextl
680 rem printen beoordeling
690 print"[op]","[3xrechts]"e,"[rechts]"
    f,"[4xrechts]"c
700 forl=1to500:next:form=1to10:getx$
    :nextm
710 loop
720 ifc<tthent=c:rem snelste oplossin
    g
730 w=w+1:x=5:goto760:rem aantal keer
     geraden
```

INGS - C-16 LISTINGS - C-16 LISTINGS - C-16 LISTINGS

```
740 forl=1to3:sound1,900,5:form=1to10
    0:nextm:nextl
750 rem uitvoeringen op beeldscherm
760 print"[HOME][19xneer]";chr$(27)+chr$
    (84)
770 onxgoto790,810,830,860,900,960
780 rem spelregel- of typefouten
790 print"[rechts]je hebt 2 dezelfde
    getallen ingetikt."
800 print"[rechts]dat mag niet.":goto
    1010
810 print"[rechts]je hebt geen getal
    ingetikt"
820 print"[rechts]tussen 0 en 10.":goto
    1010
830 print"[rechts]je hebt de nul inge
    tikt."
840 print"[rechts]die doet niet mee i
    n dit spel.":goto1010
850 rem waarschuwing laatste beurt
860 sound1,800,10:form=1to200:nextm:sound
    1,700,20
870 print"[rechts]let op! je hebt nog
     maar 1 beurt."
880 print"[rechts]denk dus goed na.":
    goto1020
890 rem indien geraden
900 forl=1to4:form=700to900+l*28step1
    5:sound2,m,1:nextm:nextl
910 printtab(8);d$
920 printtab(8);"[RVS-aan] geraden i
    n";c;"[links][2xspatie]":printtab(
    23);"[op][RVS-aan]beurten "
930 printtab(8);d$
940 goto1050
950 rem indien niet geraden
960 forl=1015to1step-20:sound1,l,1:next
    l
970 print"[rechts]helaas, je hebt het
     niet gehaald."
980 print"[rechts]de getallen waren:"
    ;a(1);a(2);a(3);a(4)
990 print"[rechts]je moet nog maar ee
    ns goed oefenen.":goto1050
1000 goto1050
1010 print"[rechts]probeer het opnieuw
    ."
1020 print"[rechts]▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬
    ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
1030 forl=1to10:getx$:nextl:return
1040 rem vragen verder spelen
1050 print"[neer][rechts]nog een spel?
     type ja of nee (j/n)."
1060 getkeyx$:ifx$<>"j"andx$<>"n"then1
    060
1070 print"[2xHOME]"
1080 ifx$="j"then260
1090 rem eindresultaat + stoppen
1100 forl=1to2:sound1,720-l*130,10:for
    m=1to100:nextm,l:sound1,230,25
1110 print"[CLR-HOME]";a$;a$;a$:color0
    ,13,1:color4,8,7:color1,4,7
1120 print"[2xneer]";tab(8);"aantal ke
    er gespeeld:";v+w
1130 print"[2xneer]";tab(8);"aantal ke
    er geraden :";w
1140 print"[2xneer]";tab(8);"aantal ke
    er gefaald :";v
1150 ift=16thenprint"[4xneer]":goto118
    0
1160 print"[2xneer][3xrechts]de snelst
    e oplossing werd gevonden"
1170 print"[neer][3xrechts]in";t;"beur
    ten."
1180 print"[2xneer]";tab(10);"[RVS-aan]
    [WIT] einde masterbrain "
1190 print"[2xneer]";a$;a$;a$
1200 printchr$(27)+chr$(76):color1,13,
    1
1210 end

*** EINDE LISTING ***
```

masterbrain c-16

regel 10	182	regel 630	75
regel 20	10	regel 640	185
regel 30	165	regel 650	207
regel 40	118	regel 660	186
regel 50	137	regel 670	206
regel 60	18	regel 680	217
regel 70	75	regel 690	116
regel 80	112	regel 700	245
regel 90	148	regel 710	236
regel 100	251	regel 720	106
regel 110	215	regel 730	171
regel 120	148	regel 740	35
regel 130	14	regel 750	235
regel 140	141	regel 760	29
regel 150	204	regel 770	248
regel 160	128	regel 780	7
regel 170	126	regel 790	250
regel 180	199	regel 800	139
regel 190	237	regel 810	151
regel 200	117	regel 820	179
regel 210	219	regel 830	177
regel 220	225	regel 840	94
regel 230	173	regel 850	178
regel 240	136	regel 860	28
regel 250	23	regel 870	55
regel 260	129	regel 880	106
regel 270	39	regel 890	60
regel 280	223	regel 900	94
regel 290	140	regel 910	64
regel 300	251	regel 920	148
regel 310	114	regel 930	64
regel 320	92	regel 940	79
regel 330	67	regel 950	108
regel 340	56	regel 960	243
regel 350	217	regel 970	171
regel 360	19	regel 980	132
regel 370	162	regel 990	196
regel 380	144	regel 1000	79
regel 390	136	regel 1010	63
regel 400	252	regel 1020	242
regel 410	22	regel 1030	220
regel 420	135	regel 1040	225
regel 430	76	regel 1050	186
regel 440	1	regel 1060	216
regel 450	58	regel 1070	3
regel 460	64	regel 1080	134
regel 470	222	regel 1090	184
regel 480	215	regel 1100	142
regel 490	48	regel 1110	191
regel 500	254	regel 1120	170
regel 510	147	regel 1130	87
regel 520	139	regel 1140	68
regel 530	220	regel 1150	77
regel 540	12	regel 1160	67
regel 550	138	regel 1170	45
regel 560	94	regel 1180	154
regel 570	169	regel 1190	223
regel 580	177	regel 1200	136
regel 590	205	regel 1210	128
regel 600	60		
regel 610	73		
regel 620	137	ready.	

INGS - C-16 LISTINGS - C-16 LISTINGS - C-16 LISTINGS

Cijfers

U weet waarschijnlijk wel dat ons cijferschrift van Arabische oorsprong is. Ook bijna iedereen weet dat er ook nog zoiets als een Romeins cijferschrift bestaat. Deze cijfers worden tegenwoordig ook nog gebruikt, u kunt ze bijvoorbeeld bij een coprightstatement aantreffen. Vanaf nu kunt u dus met uw Plus IV en C-XVI het een en ander omrekenen.

```
 0 rem romeinse cijfers
 1 rem "voor de C-XVI en de plus IV"
 2 data1000,m,500,d,100,c,50,1,10,x,
   5,v,1,i:forf=1to7:readr(f),r$(f):
   next:dimg(21)
 3 print"[CLR-HOME][neer] kies uit:[3xsp
   atie](0=stoppen)
 4 print"[neer] a. arabisch - romein
   s":print" r. romeins - arabisch
 5 getkeyg$:ifg$="a"then14
 6 ifg$<>"r"then5
 7 a=0:forf=1to20:g(f)=0:next:input"
   [neer]romeins getal";g$:ifg$="0"then3
 8 iflen(g$)>20then7
 9 forf=1tolen(g$):t$=mid$(g$,f,1):for
   g=1to7:ift$<>r$(g)thennext:goto7
10 g(f)=g:nextf:g(f)=8:print"arabisc
   h getal:";:forf=1tolen(g$)
11 forg=f+1tolen(g$):ifg(f)<=g(g)then
   nextg:a=a+r(g(f)):goto13
12 b=0:forh=ftog-1:b=b+r(g(h)):nexth
   :p=r(g(g))-b:a=a+p:f=g
13 nextf:printa:goto7
14 printr$:r$="":input"[neer]arabisc
   h getal";g:ifg>10000then14
15 ifg=0then3
16 eg=g:print"romeins getal: ";:forf
   =1to7:ifg>1000then21
17 ifg=3org=8then21
18 forp=2to7:ifg<=r(p)thennextp:goto21
19 p=p-1:ifg=r(p)thenr$=r$+r$(p):goto14
20 fori=p+1to7:forj=1to2:ifr(p)-j*r(
   i)=gthen25:elsenextj,i
21 a=int(g/r(f)):ifa=0then24
22 ifa>3theng=eg:goto26
23 forh=1toa:r$=r$+r$(f):next:g=g-a*r(f)
24 nextf:goto14
25 forh=1toj:r$=r$+r$(i):next:r$=r$+
   r$(p):ift=0then14:elseg=o:t=0:goto18
26 ift=1theng=og:q=int(g/100)*100:t=
   2:o=g-q:g=q:goto18:elser$=""
27 ifg>1000thenr$=r$+"m":g=g-1000:goto27
28 q=int(g/10)*10:og=g:t=1:o=g-q:g=q
   :goto18
29 rem
30 rem door durk bouma
31 rem uit nijland / 05156-550

*** EINDE LISTING ***
```

cijfers

regel 0	247	regel 11	213	regel 22	7
regel 1	8	regel 12	19	regel 23	215
regel 2	2	regel 13	214	regel 24	240
regel 3	158	regel 14	175	regel 25	123
regel 4	224	regel 15	142	regel 26	24
regel 5	120	regel 16	226	regel 27	214
regel 6	204	regel 17	162	regel 28	211
regel 7	55	regel 18	147	regel 29	143
regel 8	251	regel 19	13	regel 30	109
regel 9	24	regel 20	220	regel 31	120
regel 10	24	regel 21	203		

ready.

Kleinkunst

Het gaat goed met de Commodore-128. Het gevolg hiervan is dat we inmiddels enige programma's voor deze machine ontvingen. Daar we het **Visilistprogramma** nog niet voor de 128 hebben aangepast plaatsen we dus de originele listing en daar we ook nog geen checksum programma voor de nieuwste commodoretelg hebben zult u de listing dus met uiterste zorgvuldigheid moeten overtikken.
Het programma kleinkunst demonstreert de krachtige basic van deze machine. Met weinig regels krijgt u al snel een schitterende serie grafische beelden.

```
1 REM KLEINKUNST / COMMODORE-128
2 REM DOOR J VAN LEUSEN
3 REM UIT ALKMAAR / 072-121530
4 REM 6/
10 PRINTCHR$(147):COLOR0,1:COLOR1,2:COLOR4,1:GRAPHIC1,1
20 GOSUB450
30 FORR=180TO0STEP-10
40 CIRCLE1,(R/2+100)/1.2,(R/3+80)/1.2,R/1.2,R/2/1.2,,,R,10
50 NEXTR
60 GETA$:IFA$=""THEN60
70 SCNCLR
80 GOSUB450
90 FORR=240TO0STEP-10
100 CIRCLE1,(R+80)/2,R/2,R/2,,,,R/2,120
110 NEXTR
120 GETA$:IFA$=""THEN120
130 SCNCLR
140 GOSUB450
150 A=0:B=320:S=10
160 FORX=ATOBSTEPS
170 DRAW1,0,0TOX,200
180 IFX=320THENS=-S:A=315:B=5:GOTO160
190 IFX=5THENS=-S:A=2:B=322:GOTO160
200 IFX=322THENS=-S:A=317:B=7:GOTO160
210 NEXTX
220 GETA$:IFA$=""THEN220
230 SCNCLR
240 GOSUB450
250 FORK=90TO5STEP-5
260 L=L+5
270 CIRCLE1,165,105,K,,K-5,270-L,,90
280 IFK=0THEN300
290 NEXTK
300 K=0:L=0
310 FORK=90TO5STEP-5
320 L=L+5
330 CIRCLE1,155,95,K,,270-L,K-5,,90
340 IFK=0THEN360
350 NEXTK
360 GETA$:IFA$=""THEN360
370 SCNCLR
380 GOSUB450
390 FORY=5TO200STEP5
400 BOX1,0,Y-1,Y,200
410 NEXTY
420 GETA$:IFA$=""THEN420
430 GRAPHIC0,1
440 END
450 CHAR1,29,0,"GRAPHIC-ART",1
460 RETURN

READY.
```

Bijtijds

Als u op programmatechnisch gebied bij de tijd wilt blijven dan moet u absoluut deze listing even intikken. Binnen een paar minuten heeft u er dan een fraai programma voor uw 128 bij. Als u het 'Runt' weet u wel hoe laat het is.

```
1 REM BIJTIJDS / COMMODORE-128
2 REM DOOR J VAN LEUSEN
3 REM UIT ALKMAAR / 072-121530
4 REM 6/
10 INPUT"AANTAL MINUTEN (0-59)";N:M=N*6
20 INPUT"AANTAL UREN (0-11)";U:O=U*30+M/12
30 COLOR0,1:COLOR4,1:COLOR1,2:GRAPHIC1,1:PRINTCHR$(147)
40 FORSA=0TO330STEP30
50 X=99*COS((90-SA)*0.0174533)
60 Y=-X*TAN((90-SA)*0.0174533)
70 A=(X/99)*90:B=-A*TAN((90-SA)*0.0174533)
80 DRAW1,A+160,B+100TOX+160,Y+100
90 NEXTSA
100 GOTO290
110 FORSE=0TO354STEP6
120 A=87*COS((90-SE)*0.0174533)
130 B=-A*TAN((90-SE)*0.0174533)
140 G=10*COS((90-SE)*0.0174533)
150 H=-G*TAN((90-SE)*0.0174533)
160 DRAW1,G+160,H+100TOA+160,B+100
170 IFSE=0THEN190
180 GOSUB390
```

C-128 LISTINGS - C-128 LISTINGS - C-128 LISTINGS - C-128 LISTIN

```
190 IFA=CANDB=DTHEN210
200 DRAW0,G+160,H+100TOA+160,B+100
210 NEXTSE
220 DRAW0,159,100TOC+159,D+100
230 DRAW0,160,100TOC+160,D+100
240 DRAW0,161,100TOC+161,D+100
250 DRAW0,159,100TOE+159,F+100
260 DRAW0,160,100TOE+160,F+100
270 DRAW0,161,100TOE+161,F+100
280 M=M+6:O=O+0.5
290 C=87*COS((90-M)*0.0174533)
300 D=-C*TAN((90-M)*0.0174533)
310 E=65*COS((90-O)*0.0174533)
320 F=-E*TAN((90-O)*0.0174533)
330 DRAW1,160,100TOC+160,D+100
340 DRAW1,161,100TOC+161,D+100
350 DRAW1,159,100TOE+159,F+100
360 DRAW1,160,100TOE+160,F+100
370 DRAW1,161,100TOE+161,F+100
380 GOTO110
390 PRINT"DEZE SUBROUTINE IS INGEVOERD O
M DE SECONDEWIJZER VAN DE KLOK TE VERTRA
GEN. DE SLEEPFUNCTIE WERKT ALLEEN MET HE
LE SECONDEN EN IS ONBRUIKBAAR OMDAT DAN
400 PRINT"DE SECONDEWIJZER TE LANG WACHT
. MET DEZE SUBROUTINE IS DEZE KLOK TE 'I
JKEN'.
    "
410 PRINT
420 RETURN

READY.
```

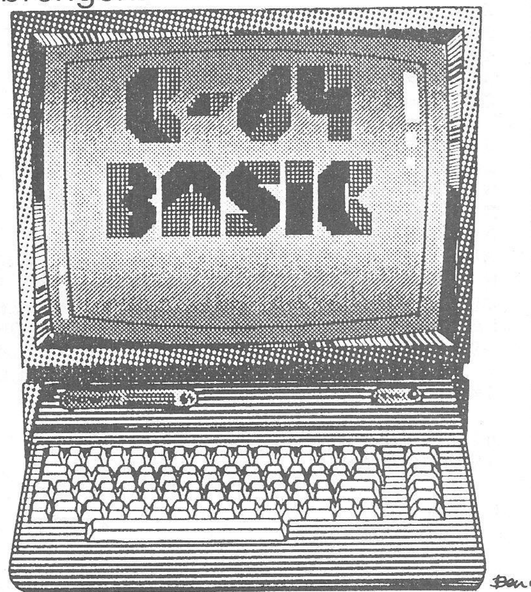
INZENDVOORWAARDEN

Programma's en artikelen kunnen aan de redactie van MSX-Info, PB 112, 1260 AC te Blaricum met vermelding REDACTIE MSX, worden ingezonden en worden eventueel geplaatst, waarbij de inzender een door de redactie te bepalen vergoeding ontvangt. Vermeld altijd naam, adres en telefoonnummer. Listings worden alleen behandeld, indien men ze ook op cassette of diskette instuurt. SAC behoudt zich het recht voor, er redactionele wijzigingen in aan te brengen. Door inzending accepteert men, dat bij plaatsing alle publikatierechten, ook in electronische vorm, op de uitgever van het blad overgaan en dat men het blad vrijwaart van aanspraken van derden, bijvoorbeeld omdat het programma eigenlijk van een ander is of bv. deels of geheel uit een ander blad is overgenomen.
Het overnemen van listings door lezers is toegestaan, maar verkoop in welke vorm dan ook, van die listings, aan derden is voorbehouden aan SAC.

In deze serie artikelen wordt stap voor stap een basis gelegd om - al programmerend - zelf met Basic en de Commodore op pad te kunnen. In de derde aflevering gaat Jan Bodzinga dieper in op het verwerken van variabele gegevens in een programma. Daarbij komt naast het toetsenbord ook de manier ter sprake, om gegevens in het programma zèlf te kunnen bewaren. De 'cursus' vereist geen speciale vooropleiding, zodat iedereen die zich voor het maken van Basic-programma's interesseert kan meedoen. De computer die ervoor nodig is, kan variëren van een VIC-20, via de C16 en C64 tot de C-128. Naast de uitleg van de Basic-opdrachten en -commando's worden allerlei praktische en handige tips besproken. Aan de hand van de voorbeelden kan de lezer zelf aan de slag om het geleerde in praktijk te brengen.

BASIC BASIS

Deel 3



In de vorige afleveringen van deze cursus hebben we kennis gemaakt met de meest elementaire opdrachten ofwel commando's waarmee een Basic programma kan worden opgebouwd. De laatste keer zijn we blijven steken middenin een programma waarmee de temperaturen kunnen worden omgerekend van graden Celcius naar Fahrenheit. Dit is een vrij eenvoudig programma, maar juist daardoor is zo'n voorbeeld goed te gebruiken om wat simpele Basic-aardigheden te illustreren.

Omdat er nog wel het één en ander over een soortgelijk programma te vertellen is, nemen we iets dergelijks nog een keer onder de loep. Voor de variatie hebben we de listing uit het vorige blad wat veranderd, zodat er nu graden Fahrenheit moeten worden ingevoerd, die dan door de Commodore worden omgerekend naar graden Celcius. Bovendien zullen we daarna dit programma een nog andere vorm geven, waarbij we niet zelf op het toetsenbord gegevens hoeven in te typen, maar dat vooraf kunnen doen door de diverse temperaturen die we berekend willen zien in het programma zelf opnemen.

Maar eerst de aangepaste toetsenbord-versie:

```
10 REM Fahrenheit / Celcius
20 REM auteur : P.Bruin 860220
30 REM
40 REM Temperaturen omrekenen
50 REM van Fahrenheit naar Celsius
60 REM
70 REM Gebruikte variabelen :
80 CE = 0 :REM Celsius
90 FH = 0 :REM Fahrenheit
100 REM Begin programma
110 PRINT CHR$(147)
120 PRINT "Graden Fahrenheit ";
130 INPUT FH
140 CE = (FH - 32) / 9 * 5
150 PRINT
160 PRINT FH; " FAHRENHEIT ="; CE; "
    Celcius"
170 PRINT
180 END
```

Als we dit programma letterlijk overtypen en daarna het commando LIST geven, moet de lijst van programma-regels er net zo uitzien als hierboven. Verschillen tussen beide listings kunnen met hulp van de cursortoetsen worden verbeterd. Runnen we dit programma, dan zullen alle getallen die we invoeren als graden Fahrenheit door dit programma (en de Commodore) naar Celcius worden omgerekend en op het scherm worden geprint. We zullen bij het uitvoeren van dit voorbeeld meteen zien, dat er bijzonder nauwkeurig door de Commodore kan worden gerekend, want de meeste uitkomsten in graden Celcius blijken nogal wat cijfers achter de punt te hebben. Dat komt omdat de deling door 9 vaak op een repeterende breuk uitkomt. De computer gaat hiermee trouw aan de slag en print een zeer uitvoerige lijst met decimalen op het scherm, zodat we op de miljoenste graad nauwkeurig de waarde in Celcius te weten komen, hoewel we ook daar nu niet meteen om zitten te springen. Op den duur zul je zien dat al deze dingen door de programmeur in de hand te houden zijn, voor het moment kunnen we volstaan met de manier waarop de uitkomst van de berekening op het scherm wordt gezet.

Vergelijking

Als we dit programma vergelijken met de listing van het voorbeeld uit de vorige Commodore-Info, dan zit het wezenlijke verschil tussen beide voorbeelden in de regels 130 en 140.

Dit komt erop neer, dat bij de INPUT in regel 130 nu het ingetoetste getal wordt toegekend aan de variabele FH (voor Fahrenheit) waar het eerste voorbeeld deze waarde aan de variabele CE (Celcius) gaf.

Ook de berekening uit regel 140 is daarom aangepast. Het is een goede oefening beide berekeningen naast elkaar te leggen en te vergelijken met de formule uit de natuurkunde, waarmee vroeger op school de sommetjes werden gemaakt. Deze formule ziet er zo uit:

°Fahrenheit = 5/9 x °Celsius + 32

Deze rekensom is in Basic vertaald in regel 140. In het programma Celsius/Fahrenheit stond er : 140 FH = (CE/5)* 9 + 32, terwijl de rekenformule is omgekeerd in ons tweede voorbeeld.

Als je de twee naast elkaar legt en vergelijkt met de natuurkundeformule krijg je een indruk, hoe eenvoudig de vertaling naar een Basic-programma in feite in z'n werk gaat. En daarmee ben je al een eind gevorderd in het programmeren. Want het vertalen naar Basic-commando's is in feite niet veel anders dan je vooraf afvragen wat er moet gebeuren en alle berekeningen en input op een rij zetten, waarna het programma in Basic kan worden geschreven.

Nogmaals INPUT

De kern van dit programma-voorbeeld schuilt in het INPUT- commando op regel 130. De Commodore stopt met z'n programma, totdat de gebruiker een getal heeft ingetypt op het toetsenbord en daarna door het indrukken van de RETURN-toest heeft gemeld klaar te zijn met de invoer. De ingetypte temperatuur wordt toegekend aan de Basic-variabele FH. Dit is de eenvoudigste manier om Basic-programma's interactief te maken met de gebruiker. Het programma vraagt de gebruiker een waarde in te toetsen, die de temperatuur in Fahrenheit aangeeft. Maar daar weet de Commodore natuurlijk weinig van. Het enige wat hij doet is het opbergen van de ingetypte waarde in de het laatje van de variabele met de naam FH. Doe je dit laatje weer open, dan vind je daar de waarde van de variabele. Wil je een andere inhoud in het laatje stoppen, dan geef je simpel weer een commando in de vorm **FH = 32**. De oude inhoud van variabele FH wordt daarmee weggegooid en vanaf nu zit de waarde 32 in het laatje van variabele FH.

Zo ook in regel 130. De gebruiker typt hier b.v. 38, het was immers 38 °Fahrenheit en deze '38' wordt als waarde opgeborgen in de la met de naam FH.

In regel 140 wordt dan netjes berekend hoeveel 38 °Fahrenheit in °Celsius wordt. Als je even rekent en de 'formule' van regel 140 beter bekijkt, dan zul je zien, dat er feitelijk weinig verschil zit tussen de natuurkunde-formule en de computer-berekening. Wat er gebeurt is dat de waarde van de variabele FH wordt ingevuld voor de FH in regel 140, dan wordt de berekening gemaakt en de uitkomst wordt in de la van variabel CE gestopt. Maar het leuke is, dat ook de variabele FH nog de waarde 38 heeft. Er is voor de berekening als het ware alleen maar even in het laatje FH gekeken wat er in zat, zonder de inhoud te verwijderen. We kunnen nog eindeloos verder met FH. Variabele CE had waarde 0, die we toegekend hebben in regel 80, maar krijgt nu de waarde 3.333..... voor °Celsius.

Dat dit allemaal juist is kunnen we kontroleren door regel 160. Het printkommando bestaat hier feitelijk uit 4 verschillende opdrachten. De eerste is het printen van de waarde van FH (was 38) daarna komt een ';' die betekent dat we op dezelfde regel ook de rest moeten printen, dan krijgen we de tekst " FAHRENHEIT = ", die wordt letterlijk geprint, daarna komt de waarde van CE (dus 3.33...) op het scherm, met daarachter de tekst " CELSIUS". Het hele programma is hiermee uitgevoerd, behalve de laatste regel, 180 END. In feite spreekt dit commando voor zichzelf, het geeft aan dat het Basic-programma met deze regel is afgelopen. De Commodore hoeft niet verder te zoeken, of er nog meer regels zijn, want ook al waren die er, dan nog wordt met END de verwerking afgebroken.

> **END**
>
> *Het gebruik van END om aan te geven dat het programma hier moet worden gestopt, is niet direkt nodig, want als er geen regels meer zijn, dan valt er voor de Commodore weinig meer te verwerken, dus wordt er vanzelf gestopt en komt de computer terug met READY. Wel kan ik END voor iedereen die een betere struktuur in z'n programma's wil brengen aanbevelen, want anderen die later een listing van zo'n programma bekijken, kunnen dan in een oogwenk zien waar het programma begint en waar het ophoudt. Overigens mag je de opdracht END zoveel en zovaak in je programma zetten als je wilt. De Commodore stopt echter meteen bij het tegenkomen van een END.*

RUN

Het voorbeeldprogramma om de temperaturen te berekenen kan niet zoveel doen, want voor iedere keer dat er een temperatuur moet worden ingevoerd, moet het programma opnieuw worden geRUNd. En dat is natuurlijk niet de beste manier om met zo'n programma om te gaan. Dit is echter gemakkelijk te verhelpen, want we kunnen dat door de Commodore zelf laten doen. Dat gebeurt door in regel 180 het commando **END** te vervangen door **RUN**. Geven we nu de opdracht RUN dan zal ons voorbeeld worden uitgevoerd, waarbij na het ingeven van de temperatuur onmiddellijk het scherm weer wordt schoongemaakt, zonder dat we tijd hebben om de geprinte uitkomst (PRINT-opdracht regel 160) te kunnen lezen. In regel 180 komt de Commodore het commando RUN tegen, waarop meteen weer gestart wordt met het begin van het programma. Één van de eerste commando's in het programma is PRINT CHR$(147), (regel 110). Het scherm wordt weer schoongemaakt. Hoewel we dus op deze manier zovaak we maar willen een nieuwe temperatuur kunnen invoeren bij het INPUT-commando, helpt de verandering van END naar RUN in 180 ons weinig, omdat we het resultaat niet op het scherm kunnen zien. Maar ook daar is (natuurlijk) een oplossing voor. Als eerste kunnen we simpel regel 110 verwijderen uit het programma. Dat wordt bij het opnieuw runnen het scherm niet meer schoongemaakt, dus kunnen we de oude tekst op het scherm blijven lezen, terwijl de Commodore voor de tweede run alweer op een INPUT staat te wachten. Je begrijpt dat dit niet de beste oplossing is, want het commando PRINT CHR$(147) bestaat natuurlijk niet voor niets.

Een veel betere oplossing is het gebruiken van het commando **GOTO**. Daarvoor moet regel 180 worden veranderd in 180 GOTO 120. Als dit is gedaan, en het programmaatje wordt weer gerund, dan wordt het scherm eerst schoongemaakt, waarna steeds de input en de uitkomst van de berekening op het scherm verschijnt. Pas als de tekst na een paar keer invoeren onderaan het scherm is gekomen, zal de bovenste regel van het scherm verdwijnen. Tot op dat moment kunnen we de omrekening blijven lezen.

GOTO

Zoals je uit het voorgaande hebt begrepen, kunnen we het commando GOTO gebruiken om de verwerking van een programma naar een ander punt in de listing te sturen. GOTO is net als alle andere Basic-woorden van het engels afgeleid. Als je dit woord zou vertalen dan wordt het zoiets als GA NAAR (regelnr). Dit is zeker niet de moeilijkst te onthouden Basic-opdracht. Hoe gaat GOTO eigenlijk in z'n werk ? Zoals de naam al doet vermoeden geven we hiermee de Commodore opdracht naar een ander regelnummer te gaan, dan waar de uitvoer van een programma normaal mee zou worden vervolgd.

Een bijzonder gemakkelijke opdracht om te gebruiken, maar tevens de meest gehate bij de èchte programmeurs, want door een veelvuldig gebruik van GOTO bereiken we de zo genoemde spaghetti structuur, waarin de Commodore van A naar Q naar C naar X wordt verwezen, vaak zelfs zonder enig aanwijsbaar nut. Anders gezegd,

de opdracht GOTO geeft de Basic-programmeur de gelegenheid om zonder enige zichtbare struktuur zijn programma's te kunnen schrijven.
Op dit moment maakt het ons niet uit wat de geleerden van GOTO vinden, we hebben in het voorbeeld een prima manier gevonden om de uitvoer van het programma weer naar het begin van de INPUT te kunnen laten springen, zonder dat we daar op het toetsenbord ook maar iets voor hoeven te doen.
GOTO kan alleen maar in een Basic programma voorkomen, als achter het commando een geldig regelnummer staat. GOTO 109 geeft in ons geval de foutmelding **'Undefined line number in 180'**. In 't nederlands wil dat zoveel zeggen als het regelnummer dat in regel 180 staat zit niet in dit programma. Verder kun je met hulp van GOTO alle kanten op. Zelfs een regel als **180 GOTO 180** is mogelijk. Bekijk zelf maar eens wat er dan gebeurt in het programma. Als je een programma op de Commodore wilt onderbreken, kan dat altijd door op de CTRL-toets samen met de STOP-toets te drukken!

Vaste gegevens

We gaan door met ons temperatuurprogramma. Dat wordt nu zo veranderd, dat we in het programma zelf de gegevens zetten, waarmee de berekening moet worden gemaakt. Het programma moet er zo uit komen te zien:

```
 10 REM Fahrenheit / Celcius
 20 REM auteur : P.Bruin 880218
 30 REM
 40 REM Temperaturen omrekenen
 50 REM van Fahrenheit naar Celsius
 60 REM vauit DATA-regels
 70 REM Gebruikte variabelen :
 80 CE = 0    :REM Celsius
 90 FH = 0    :REM Fahrenheit
100 REM Begin programma
110 PRINT CHR$(147)
120 READ FH     :REM lezen temp
130 IF FH > 999 THEN END
140 CE = (FH - 32) / 9 * 5
150 PRINT
160 PRINT FH; " FAHRENHEIT ="; CE; "
    Celcius"
170 PRINT
180 GOTO 120
190 DATA 0, 5, 10, 15, 20, 30
200 DATA 100,200,50,23.5,45,2
210 DATA -3,-5,70,-77,1000
```

De meeste regels zijn gelijk gebleven aan het eerste programma uit deze les. Alleen bij regel 60 is een extra opmerking achter het REM-commando neergezet. Verder is de kern van het programma aangepast en zijn er aan het einde drie regels met getallen bijgekomen. Bovendien heeft regel 180 de besproken vorm met een GOTO opdracht gekregen. Dat deze hier echt op z'n plaats is, wordt straks duidelijk.
Wat zijn de veranderingen ? Als eerste hebben we nu geen INPUT- commando meer in het programma. Dat betekent dat de communicatie tussen gebruiker en Commodore tijdens het uitvoeren niet mogelijk is. Runnen we echter dit voorbeeld, dan komen er toch de nodige uitkomsten op het scherm. Rara hoe kan dat. Welnu, het programma is niet voor niets veranderd, en daar komen die veranderingen vandaan. In plaats van de INPUT hebben we nu een **READ**- opdracht in de listing gezet. Dit commando zorgt er net als INPUT voor, dat gegevens worden gelezen en in een variabele worden geplaatst. De waarden die er gelezen worden komen nu niet van het toetsenbord, maar zitten in het programma zelf. Je kunt ze vinden in de regels 190 tot 210, die alle worden voorafgegaan door het commando **DATA**. Tijdens het schrijven van de programmare-

gels hebben we er dus voor gezorgd dat de gegevens al aanwezig zijn.

READ & DATA

De commando's DATA en READ zijn erg nauw met elkaar verbonden. Zonder DATA kan READ niet worden gebruikt, terwijl DATA zonder een READ-commando niets uitricht in een programma. Zoals je kunt zien bij het runnen van dit voorbeeld worden de getallen die in de DATA-regels staan stuk voor stuk aan de variabele FH toegekend, waarna de berekening uit regel 140 erop wordt losgelaten. De uitkomst wordt geprint en een nieuwe variabele wordt via GOTO 120 door de READ in regel 120 gelezen. Je kunt nu net zoveel DATA-getallen in het programma zetten als je zelf wilt, als je er maar voor zorgt, dat het laatste DATA- getal groter is dan 999. Want in regel 130 wordt getest via een IF-commando, of de waarde die met READ in FH is gezet niet groter is dan 999. Is dit wel het geval, dan wordt de END-opdracht die je daar ziet, uitgevoerd en zal het programma de uitvoer stoppen.

De mogelijkheden van IF en THEN zullen we later bespreken, we beperken ons hier tot de DATA en READ opdrachten. De volgorde waarin de DATA-getallen worden gelezen hangt af van de volgorde waarin ze in het programma voorkomen. Bij het uitvoeren van het READ-commando wordt door de Commodore gezocht naar de DATA-regel met het laagste regelnummer. Die plaats wordt onthouden en van die regel wordt het eerste nummer gelezen en verwerkt. Bij de volgende keer, dat READ wordt uitgevoerd, via de GOTO uit regel 180 komen we weer bij regel 120 terecht, wordt dan het volgende getal uit de DATA-regel gelezen en zo verder totdat we een getal tegenkomen dat groter is dan 999.

Verwijderen we regel 130, dan zal het programma alles doen zoals het hoort, behalve de 'END- test' waardoor we de melding op het scherm krijgen '**OUT OF DATA IN 120**'. Daarmee laat de computer ons weten, dat er een READ-commando moet worden uitgevoerd, maar dat er geen DATA meer in het programma voorkomt die nog niet is gelezen. Dus ook zonder regel 130 zal het programma uit zichzelf stoppen, maar dan wel met een foutmelding. En daarvoor zijn we niet aan het programmeren. Vandaar de test op regel 130. We kunnen deze test zo veranderen, dat bij het lezen van het DATA-getal 1000 in plaats van END het commando **RESTORE** wordt gezet. Daardoor wordt de volgende maal dat READ wordt uitgevoerd, de wijzer naar het DATA-getal dat aan de beurt is, weer terug gezet naar het begin van de eerste DATA-regel (190) in het programma. Op die manier zal het programma net zolang doorrunnen tot we op RUN/STOP drukken.

We kunnen met hulp van DATA en READ allerlei informatie min of meer vast in een Basic-programma opbergen, terwijl aan de andere kant de DATA-gegevens vrij gemakkelijk zijn te veranderen. Het is natuurlijk niet de meest handige manier om in een programma gegevens te schrijven, maar voor sommige vaste zaken voor een bepaalde programma-versie kan het worden gebruikt. Denk hierbij maar eens aan de dagen van de week. Als we ze in DATA-regels wegschrijven en het programma moet worden vertaald voor b.v. franstalige gebruikers, dan is het veel gemakkelijker om een paar DATA-regels in het programma aan te passen, dan in de hele listing te moeten zoeken naar de printregels waarin dan de tekst in het frans moet worden vertaald. Ook programma's die gebruik maken van grote hoeveelheden **vaste** gegevens die door de gebruiker tijdens het runnen kunnen worden gecombineerd, kunnen bijzonder goed uit de weg met DATA en READ. Ik denk hier aan een programma als de gedichtengenerator, waarin alle afzonderlijke dichtregels als DATA worden weggezet. De 'dichter' krijgt dan de keuze uit de regels, waarbij de DATA als bron dient om er een uniek geheel van te maken.

Ons voorbeeld kan nu eigenlijk heel snel worden veranderd in het tegenovergestelde van wat er nu gebeurt, terwijl de data intact blijft. Daarvoor hoeft alleen regel 140 maar te worden vervangen door 140 FH = (CE/5) * 9 + 32, terwijl in de READ-opdracht in plaats van de variabele FH nu CE moet komen te staan. Als deze wijzigingen zijn aangebracht kunnen we precies dezelfde DATA- getallen gebruiken, maar nu zijn het 07Celsius-waarden.

Een leuke oefening om zelf eens mee aan de slag te gaan.

Even herhalen

We hebben tot hier al een aardige hoeveelheid (nieuwe) Basic-opdrachten te verwerken gekregen. Het kan daarom geen kwaad de belangrijkste nog even op een rij te zetten, waarmee we ons geheugen een beetje kunnen opfrissen.

LIST

LIST wordt gebruikt om een in de Commodore aanwezig Basic- programma op het scherm te kunnen afdrukken. Er bestaan diverse mogelijkheden bij dit commando. LIST <beginregel> - <eindregel>.

De parameters voor zowel begin- als eindregel kunnen worden weggelaten waardoor respectievelijk van het begin tot de eindregel of vanaf de beginregel tot het einde van het programma de listing verschijnt. LIST zonder parameters list het gehele programma. De listing is te stoppen door op de BREAK-toets te drukken.

REM

REM wordt gebruikt om opmerkingen over het programma of anderszins in het programma te kunnen zetten. De gegevens na een REM-commando worden niet door de Commodore uitgevoerd. REM kan ook worden gebruikt midden in een Basic regel, gescheiden door een ':' van de voorgaande opdrachten. Er zijn echter Commodore-versies bekend, waarbij het gebruik op deze manier in een DATA-regel problemen oplevert. De Commodore beschouwt dan ook de REM als normale DATA. Voorzichtigheid is daar dus geboden.

LET

*Het LET commando wordt gebruikt om een waarde aan een Basic variabele toe te kennen. Het is een optioneel commando, wat betekent dat het net zo goed kan worden weggelaten, zonder dat de werking van de Basic interpreter hierdoor wordt verstoord. De funktie : AA = 25 * 12 is dus identiek aan LET AA = 25 * 12*

PRINT CHR$(147)

PRINT CHR$(147) wordt gebruikt om in programmeermode het scherm te kunnen schoonmaken. De cursor gaat daardoor terug naar de beginpositie van het scherm, linksboven op regel 1, kolom 1. Het gehele scherm wordt tevens leeggemaakt. In direct mode kan dit ook gedaan worden door het indrukken van de SHIFT-toets samen met de CLR/HOME toets. We kunnen dit ook in een programma verwerken door te typen PRINT "SHIFT + CLR/HOME*f*". In ons voorbeeld hebben we voor de duidelijkheid de ASCII-waarde van deze toetsencompbinatie (147) genomen.

INPUT

Het INPUT-commando is hier nog maar kort aan de orde geweest. Het dient om interactie teweeg te brengen tussen Basic- programma en gebruiker. Met INPUT kunnen gegevens door de gebruiker (via het toetsenbord) aan de Commodore worden toegevoerd. Het INPUT commando zorgt dat deze waarden worden toegekend aan de variabele die daarvoor door de programmeur is aangewezen.

Een nog niet besproekn mogelijkheid is het opnemen van een te printen tekst in het INPUT-commando. Dus bv.:
INPUT "Hoeveel jaar bent U "; JR
Let hierbij op de juiste leestekens en hun plek in de syntax. Als dit commando wordt uitgevoerd, zal eerst de tekst op het beeld worden gezet, waarna de Commodore zal wachten op invoer van het toetsenbord. Ook is het mogelijk om meerdere gegevens in één keer met een INPUT te kunnen lezen. Dan moeten alle te lezen variabelen in volgorde en gescheiden door komma's achter INPUT komen. Dit kan worden toegelicht met het volgende programma:

```
  5 REM TEST INPUT 2
 10 PRINT CHR$(147)
 20 INPUT "VIJF GETALLEN";A,B,C,D,E
 30 PRINT "A = "; A
 40 PRINT "B = "; B
 50 PRINT "C = "; C
 60 PRINT "D = "; D
 70 PRINT "E = "; E
 80 PRINT
100 GOTO 20
```

END
END wordt gebruikt om de computer te laten weten, dat het programma hier op een goede manier moet worden gestopt. Er kunnen zoveel END-commando's in een programma worden gebruikt als wenselijk is. Het gebruik van END komt de programmeerstruktuur ten goede, maar is (net als LET) in de meeste gevallen optioneel.

PRINT ;
Door een ; achter een PRINT opdracht te plaatsen geven we te kennen, dat de volgende te printen karakters NIET op een niewe regel, maar direkt achter de voorgaande tekst moet worden gezet. Dit geldt voor zowel alle PRINT-achtige opdrachten als de opdrachten die bestemd zijn voor de printer. Een soortgelijk gebruik is toegekend aan de , (komma) tussen PRINT-commando's.

GOTO regelnummer
Met GOTO kunnen we de Commodore naar een ander regelnummer in het programma sturen, dan de volgende regel in het programma. Daarbij moeten we er om denken, dat het regelnummer dat achter GOTO in het programma komt te staan, de z.g. **target-regel** ook werkelijk in het programma voorkomt. Het commando GOTO is een erg gemakkelijk te gebruiken hulpmiddel om de uitvoer van het programma te laten doen wat wij willen. Aan de andere kant zorgt met name deze opdracht ervoor dat een programma meer gaat lijken op spaghetti of vermicelli dan op een goed gestroomlijnd stukje software. Je zou kunnen stellen dat een echt goed Basic programma geen enkele GOTO-opdracht bevat. Ook voor de voorbeelden die we tot nu toe hebben behandeld heeft Commodore-Basic waar het GOTO betreft wel een alternatief.

DATA
DATA is een Basic-commando waarmee wordt aangegeven dat de rest van de regel bestaat uit min of meer losse gegevens waarvan het programma gebruik kan maken. De plaats van de DATA-regel in een listing is van geen enkele belang. De gegevens worden wel gelezen vanaf het laagste DATA-regelnummer, tenzij er met **RESTORE regelnummer** een ander begin wordt aangegeven. Er mogen zoveel getallen of tekst achter DATA komen te staan als de ruimte in een programmaregel toelaat. De READ-opdracht wordt gebruikt om de DATA-gegevens te lezen en daarna te gebruiken in het programma zelf. De Commodore onthoudt welk getal (of tekst) het laatst door READ is gebruikt. Wanneer in het programma opnieuw een READ wordt uitgevoerd zal het volgende DATA-item worden gebruikt. Teksten hoeven niet tussen aanhalingstekens te staan, maar kunnen natuurlijk alleen door stringvariabelen worden gelezen. Anders volgt er een 'SYNTAX-ERROR'. Alle items op een DATA-regel moeten door een komma worden gescheiden.

READ
READ leest de items uit DATA-regels en zet ze in de adressen van de variabelen die achter het READ-commando staan. Net als bij INPUT kan ook het READ-commando worden gebruikt om meerdere gegevens in één keer in te lezen. READ A,B,C,D is dus een geldig Basic-commando. READ kan alleen worden gebruikt als er DATA-regels in een programma voorkomen. Zijn die er niet, of te weinig dan volgt een '**OUT OF DATA**' fout. READ kan worden gebruikt met numerieke en tekstvariabelen. Op deze laatste soort van gegevensopslag zullen we binnenkort uitvoerig ingaan. READ begint te lezen bij het DATA-statement met het laagste regelnummer en gaat door met de hogere regels uit het programma. Wil je per sé DATA uit een bepaalde regel lezen, dan kan dit alleen in samenwerking met het RESTORE-commando.

RESTORE
RESTORE kan de teller naar het DATA-item dat aan de beurt is wijzigen. Zoals we gezien hebben wordt er door de Commodore een telling bijgehouden om bij de uitvoer van READ te kunnen bepalen welk DATA-item moet worden gelezen. Bij RESTORE wordt deze DATA-wijzer terug gezet naar de laagste (eerste) programmaregel waarin een DATA-opdracht staat. ●

Werk-Markt
In het vorig nummer meldden we, dat we als speciale service aan onze lezers ook aandacht zouden gaan geven aan mogelijkheden voor werken in de micro-automatisering.
Veel enthousiaste micro- hobbyisten zoeken naar mogelijkheden om van hun hobby ook hun beroep te maken of hun kennis op een of andere manier nuttig te gebruiken.
Hoewel de tijd tussen uitkomen van het blad en de sluiting van het volgende zeer kort is, hebben we toch al een paar reakties te melden.

Gezocht: Werk als Field-engineer of operator door computerenthousiast (man) van 38 jaar, die een cursus informatica volgde en nu de MEAO -Informatica volgt. Bolsward (05157- 4972).

Gevraagd : Programmeur voor het maken van oefenmateriaal voor revalidatie van patienten met hersenletsel. Moet helaas wel hobby blijven. Inl. Dr. L.de Vos, Afd. Revalidatie-Dijkzicht Rotterdam (010-633445 of 010-220381 s'avonds.
MCN Whiz-Kids De Microcomputer Club Nederland zoekt weer een aantal mensen. Ditmaal meer gericht op de PC toepassing. Inl. Maurice de Hond, MCN, De Klencke 6, 1083 HH Amsterdam.
Gezocht Werk met computertoepassingen door onderwijzeres (30 jr.) Bereidheid tot volgen van studie. Alphen ad Rijn.
Gevraagd Partieel leerplichtige om op de administratie van Commodore Info in Amsterdam te assisteren. Inl. J.Taverne 020-273198

Heeft u ook werk of zoekt u juist werk, schrijf ons dan even, we plaatsen deze Werk-Markt annonces gratis.

In den beginne waren er de eenvoudige en tamelijk rechtlijnige video-schietspelletjes. De "dagen van Pong" zijn echter voorbij en ook Pacman is een historische bezienswaardigheid, om maar niet te spreken van Defender en Asteroïds. De op de ruimtewezens behaalde zeges werden daarbij met de nodige botsingen en bijbehorende explosies omlijst. Als spelconsument raak je echter snel verwend door de vele mogelijkheden die de C-64/C128-machines bieden. Met eenvoudige schietgames hoeven de softwarehuizen niet meer aan te komen. Maar de grens van het grafische geweld is bijna bereikt, nu moeten de softwarehuizen echt hun best doen en vooral gecompliceerde spelletjes maken, die meer vragen dan oog/hand coördinatie. Adventures, zowel pure tekstversies als de gemengde vormen met tekst, beeld en actie, maken tegenwoordig een groot deel van het spelaanbod uit.

Digitaal avonturen

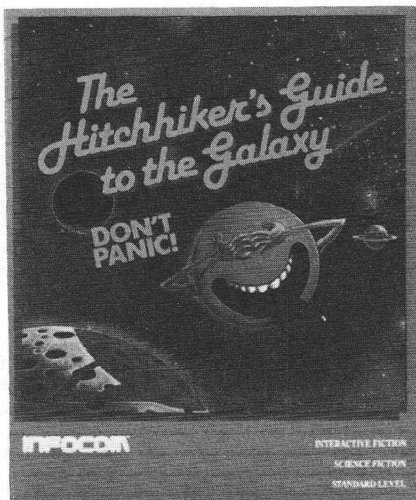
Een vlucht uit de werkelijkheid?

Voor het Engelse woord "Adventure" bestaat geen goede nederlandse vertaling. Sinds Scott Adams deze "kunst"vorm naar de microwereld overbracht is het woord een eigen leven gaan leiden. Avonturenverhaal is een vlag die de lading nauwelijks dekt en vooral voorbij gaat aan het interactieve aspect, de mogelijkheid om de loop en uitkomst van het verhaal te beinvloeden. Een adventure is meer dan een avonturenverhaal, het is een totaal nieuwe omgeving, een gedachtensprong uit de werkelijkheid en dat vooral in interactie met de gebruiker. Termen als "Interactive Fiction" of computerroman worden ook wel gebruikt om deze categorie software mee aan te duiden.

Een adventure speelt vrijwel altijd in een voor de speler ongewone omgeving of sfeer. Of hij/zij nu als "Hacker" in een vreemde computer ronddoolt, als "Ghostbuster" geesten vangt, of opwindende reizen door ruimte en tijd naar nieuwe werelden maakt, het zijn geen alledaagse omgevingen. Er is sprake van ontsnapping aan de sleur van alledag en dat wordt versterkt door de mogelijkheid, zich met de spelfiguur te vereenzelvigen. Het rollenspel-aspect is dan ook zeer belangrijk, de hoofdfiguur moet in eigenschappen en gedrag de identiteit (of gewenste identiteit) van de speler kunnen vertolken. Meestal zijn dergelijke spelfiguren macho-achtige types, maar wat te denken van een indentificatie met een morsige en onvrijwillige antiheld. Neem Heinrich Glühwein, die het prototype van de decadente "vieze man" speelt in **Hollanditis** met het hete hangijzer van de kruisraketten als probleem en de nucleaire vernietiging als afstraffing.

Het is net zo ondoenlijk om Adventures te klassificeren, als het mogelijk is de wereldliteratuur in honderd zinnen afdoend te beschrijven. Het genre breidt zich nog dagelijks uit.

Of hij/zij nu in de griezelkerk van **De Sekte** ronddoolt, als **007** May Day door Parijs achtervolgt of op vreemde werelden boze krachten bestrijdt, er is sprake van ontsnapping aan de sleur van alledag, een overstapje naar een andere wereld door zich met de spelfiguur te vereenzelvigen. De speler moet een gedachtensprong kunnen maken om zich met de desbetreffende hoofdfiguur te identificeren en het spel tot een goed einde te kunnen brengen. Zij die zich niet aan de identificatieregels houden, komen echt niet ver!



Interactie

Met name de realistische interactie met de gebruiker vormt een recente ontwikkeling bij de adventure-spelen. De ontwerpers pogen de speler zodanig bij het spel te betrekken dat hij/zij de werkelijkheid vergeet en de spelsituaties als (pseudo)realiteit ervaart.

Soorten adventures

Er zijn vele adventuretypen en -vormen. Wij zullen ons hier slechts met de belangrijkste soorten bezighouden:

▷ **Tekstadventures** zijn de oudste spelvorm. Zoals de naam al aangeeft zijn er nauwelijks of geen "graphics" aanwezig. Scott Adams bracht de eerste over van de mini- en mainframecomputers naar de hobbycomputer en is de grondlegger van het genre. Op het eerste gezicht lijkt een tekstadventure op zich saai, is dat nu alles, wat regels tekst op beeld?!

Bij het spelen van een goede tekstadventure blijkt de beperking tot letters juist een stimulans voor de fantasie.

Een tekstadventure is in feite een soort teksthoorspel, waarbij je net als op de radio of in een boek de verschillende karakterfiguren zelf dient in te vullen. Het is jouw beeld van de spelfiguren en er zijn geen tegenvallende graphicmaaksels die de illusie verstoren. En het werkt!

In Horror Hotel bijvoorbeeld beweegt de visueel niet, maar in jouw gedachten wel aanwezige dr.A.Cula zich door de duistere hotelkamers. Elke stap in zo'n onbekende kamer kan de laatste zijn. Elke aanflitsende tekst kan een noodlottige mededeling bevatten!

◊ **Doolhofadventures** lijken tegenwoordig per dertien in een dozijn te gaan. De term doolhof geeft al aan wat de speler te wachten staat: Ontsnappen uit een gangen-, grotten- of kerkerstelsel. En daarbij kan de wakkere held de nodige verrassingen, tegenslagen en aanvallen op het leven verwachten.
Het doolhof komt ook als onderdeel van grotere adventures voor. Dit verhoogt vaak de moeilijkheidsgraad.

◊ **"Graphic" of grafische adventures** drijven voornamelijk op de visualisatie van het scenario. Ook moet een groot deel van de sleutels tot de oplossing uit de begeleidende platen gehaald worden. Grafische adventures kunnen met name in de beeldplaatversie behoorlijk natuurgetrouw overkomen. Slechte adventure "graphics" kenmerken zich door zich veelvuldig repeterende tafrelen, waarin eigenlijk nauwelijks iets verandert.

◊ **Sound&Graphics-adventures** maken een weloverwogen gebruik van de SID-chips. Naast de (huiveringwekkende of schokkende) geluidseffecten vind je bij deze avonturenspelen steeds meer sfeertekenende achtergrondmuziek en spraaksynthese.

Het interactief adventure probeert de speler met grote realiteitszin in het spel op te nemen. Althans bij de betere fabrikaten. Sommige softwarehuizen noemen elk spel dat vragen stelt of opdrachten geeft interactief. Ook hier zal het gebruik van laser-disks met video informatie en CD-ROM'S nieuwe dimensies kunnen openen!

Tot slot nog een indeling naar het adventureonderwerp. Zo onderscheiden wij griezel-, politieke-, strategische-, detective-, science fiction-, tovenaar- en avonturenroman-, handels-, simulatie- psychedelische- en actieadventures. Ook is het tegenwoordig mode aan het worden, aan een adventure en het vinden van de oplossing, danwel schat, een prijsvraag te verbinden.

Hoe werkt een adventure?

Met een adventure kunnen de makers twee kanten uit: Het actiegame en het deductievespel. Bij het actiegame staat de speler meteen in het middelpunt van de schermutselingen en de elkaar snel afwisselende gebeurtenissen. Zo vertoeft de hoofpersoon in Ultima II (Aackosoft) in vreemde kastelen, op koninklijke hoven, maakt vrienden en vijanden, en voert strijd door ruimte en tijd.
Bij het deductieve spel ligt de nadruk op het vergaren van gegevens en sleutels die tot de oplossing van het spel moeten leiden. Je ziet echter steeds vaker dat adventureontwerpers een adventure opdelen in afwisselend actie- en deductieve scenes
De besturing van de hoofdpersoon stelt menige speler voor breinkrakende problemen. Het oplossen van het adventureraadsel vergt een goede observatie van de "graphics" en tekst en een flinke dosis logisch verstand. Vooral de adventures die met sleutelletters, -woorden of zinnen werken zullen vele avonden en zweetdruppeltjes kosten. Daarbij komt vaak nog het probleem van de Engelse taal. Gelukkig zijn er nederlandstalige adventures van ondermeer Radarsoft.
In de eenvoud ligt dikwijls de oplossing. Met simpele opdrachten als kijk (look), onderzoek (search/examine), open (open/unlock), oost/west/noord/zuid (east/west/north/south), help, in/uit (in/out), sleutel (key), pak (get), ontsteek lamp (light lamp), door de deur (go/through door/gate), spring (jump), inventaris (inventory) e.d. kom je al een heel eind. Verder is het onverstandig onbekende voorwerpen aan te raken, zo maar deuren te open, ergens onvoorbereid naar binnen te gaan of waarschuwingen in de wind te slaan! Dan ben je snel uitgespeeld.
Goede adventures reageren op meerdere opdrachten van dezelfde strekking. Zo dienen kijk, onderzoek en zie tot hetzelfde resultaat te leiden. In sommige moeilijke spelfasen is er slechts één verlossend woord of verlossende zin. En als je de sleutel echt niet kunt vinden kijk dan ook eens onder de deurmat of een aantal schermen terug. Wellicht is daar wat over het hoofd gezien!
Houdt altijd voor ogen dat de makers van adventures ook menselijk zijn. Probeer je in de gedachtenwereld van de spelontwerper in te leven en breng gangen- of grottenstelsels in kaart. Maak notities van nuttige aanwijzingen of opgedane ervaringen. Dan is elk adventure tot een goed einde te brengen.
Behalve het toetsenbord voor de in te typen commando's wordt ook de joystick veelvuldig gebruikt. Niet alleen voor de actiescenes maar ook voor de keuzeopties van de spelmenu's.

Een kleine adventuregreep

Het aanbod aan adventures is zo groot dat het gewoon ondoenlijk is om hen allemaal te bespreken. Daarom beperken wij ons hier tot een illustratieve keuze van nieuwe en al wat oudere adventures:

Hollanditis



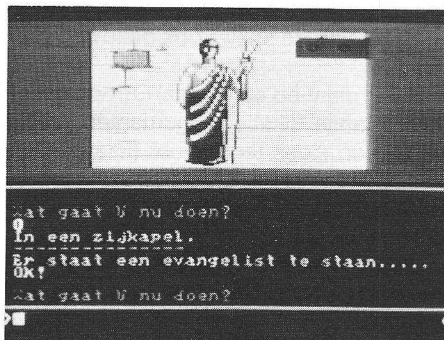
Een ongewoon adventure van Radarsoft over de politieke- en ethische kanten van het kruisrakettenvraagstuk. De antiheldfiguur, de decadente oude "vieze" man Heinrich Glühwein kan het spel zowel pro als contra spelen. Daarin blijft Hollanditis ook geheel neutraal want de publieke opinie blijft bepalend voor de uitslag. Deze uitslag kan door Glühwein beinvloed worden. Door hem gemaakte fouten werken de opzet van de speler ook daadwerkelijk tegen.
Heinrich dient op een aantal Europese- en Amerikaanse lokaties argumenten, voorwerpen en clues tegen/voor (naar keuze) de plaatsing van kruisraketten te verzamelen. Het leuke is daarbij dat deze antiheld eigenlijk helemaal niet in het politieke gebeuren geinteresseerd blijkt.
Behalve Glühwein trachten ook de even slechte KGB en CIA in de politieke modder te roeren. Daarbij gebruiken zij dezelfde praktijken. Er is geen sprake van een "Goodie-Baddy"-typering.
Ook de politieke gezindheid van de verschillende landen komt in dit politieke adventure tot uitdrukking. De Italianen zijn bijvoorbeeld heethoofdig en de Amerikanen trachten zo veel mogelijk feiten in de doofpot te stoppen. Een fraaie couleur locale.
De door de speler verzamelde clues kunnen ter beoordeling in een soort WO III-simulator gestopt worden om b/v. de publieke opinie af te tasten.
Alle beeldlokaties in Hollanditis zijn fraai gedetailleerd en vervullen meer dan alleen een achtergrondfunctie. Je zult de in de "graphics" verwerkte aanwijzingen echt nodig hebben. Ook de muziek en het geluid (o.a. rinkelende telefoons) zijn goed verzorgd. Glühwein beschikt zelfs over spraaksynthese en uit zich geïrriteerd of platvloers (hoppa).

Horror Hotel

Dit griezelspel van Radarsoft is een echt interactief tekstadventure. Wij kunnen van een soort visueel hoorspel spreken, want je moet in gedachten alle karakters en figuren zelf invullen.
Horror Hotel bestaat uit drie onderling

verbonden avonturen. Er zijn twee plots en er dient bovendien nog een vloek opgeheven te worden. En dat gaat beslist niet van een leien dakje, daar het in Transilvania gesitueerde hotel onveilig wordt gemaakt door de gruwel dr.A.Cula (hoe origineel). Een ontmoeting met deze maligne ontspoorde dokter doet het adventure nogal eens voortijdig beeindigen. En maak ook geen deuren met het opschrift niet openen toch open, want de weerwolf is dol op eigenwijze avonturiers/sters.
Het feit dat je behalve de tekst niets ziet, werkt flink griezelsfeer verhogend. Welke (noodlottige?) tekst zal op het volgende scherm aanflitsen? De maker van Horror Hotel, John Vanderaart, vindt dit adventure zelf zo leuk dat hij betreurt dat het niet door een ander geschreven is. Maar ja, als je de oplossing al weet gaat veel van de spanning verloren.

De Sekte



De Sekte (Radarsoft) is een volledig nederlands mysterieadventure. Na de verdwijning van de sekteleider Leon H. Rabburd uit de "duivelskerk" voltrok zich in het Schotse plaatsje Achnasheen een ware uittocht van de door vreemde mentale krachten beheerste bewoners. Sinds zondag 18 November 1962 is dan ook de kerk leeg en verzegeld en Achnasheen volledig ontvolkt.
U, als avonturier, wordt aangetrokken door het feit, dat met het verdwijnen van de sekteleider ook de rijkelijk gevulde kas zoek is geraakt. Mogelijk is het geheim van het sektefortuin in de door mentale krachten beheerste kerk verborgen. Betreden van de kerk is geheel voor eigen risico!
De Sekte is een echt gecombineerd teksten "graphics"-adventure. De aanwijzingen zitten zowel in de tekst als de begeleidende plaatjes verborgen. De plot is opbouwend van aard en meerdere wegen leiden tot succes. Het aardige van dit programma is dat het zowel nederlandse (lid)woorden, voorzetsels als gehele (zakelijke) zinnen als opdracht accepteert. "Kijk altaar" is bijvoorbeeld iets anders dan "Kijk onder het altaar".
Verder beschikt De Sekte over redelijk goede muziek en geluidseffecten en weet de ontwerper het maximum uit een minimum aantal KRAM-machinetaal te halen.

Hacker

Het computerkraken is van een werkelijk tot een spelavontuur geworden. Een aardige ontsnappingsmogelijkheid voor hen, die altijd al eens in een computer wilden inbreken zonder strafbare feiten te begaan. Of een nieuwe uitdaging aan echte krakers die t.g.v. de huidige strenge veiligheidsmaatregelen de pijp aan Maarten hadden gegeven.
Aan het begin van dit Activision kraakadventure maakt de speler na maandenlang vruchteloos proberen eindelijk kontakt met een vreemde computer. In de gebruiksaanwijzing staat eigenlijk niet veel meer dan hoe je het programma moet starten na "Insert disk".
Voor het echte kraken zal je al je inventiviteit en kennis nodig hebben om je door het doolhof van het vreemde computerbrein te worstelen. En wat daar, na het vinden van het juiste codewoord, niet allemaal aan het licht komt! Bij de firma Magma Ltd. blijken belangrijke documenten door industriële spionnen ontvreemd te worden. De speler probeert het spionagenetwerk te ontrafelen door naar elf buitenlandse steden (zelfs in China) af te reizen. Aldaar kan men proberen kontact met de kraakspionnen op te nemen en allerlei belangrijke zaken te verhandelen.
Hacker zit vol leuke vondsten, zoals het toespreken in de landstaal (gelukkig met Engelse ondertiteling, al is die opzettelijk gekunsteld en daardoor wel eens niet te volgen), de buitenlandse kontakten en spionagesatellieten die door het werk zoeven. De besturing verloopt via het toetsenbord en de joystick.

Elite

Het simulatie-adventure Elite (Firebird) kun je zien als een combinatie van avontuur, ruimtemonopoly en strategie (bij de ruimteslagen). In de eenvoudige huiscomputeruitvoering geven de 3D vectorlijnen- en bewegingen weliswaar een realistisch inzicht in zo'n heelaloorlog, maar zij ontnemen de speler de illusie dat het om echte ruimteschepen gaat.
De Elite-rang mag de speler zich pas toeëigenen als hij/zij zich economisch en strategisch waargemaakt heeft. En dat kost de nodige planning, een flinke dosis handelsinzicht en slagvaardigheid. Behalve als eerlijk intergalactisch handelaar kun je ook als smokkelaar of premiejager je persoonlijk fortuin doen toenemen. De achtergrond is aardig uitgewerkt, compleet met politieke sfeertekening, navigatie- en gevechtsregels, een lijst met beschrijving van de vijandelijke schepen en een handelsregister.

Spelerinteractie

Vele adventures worden interactief genoemd. Dat betekent, dat er een wisselwerking is tussen speler en machine. Het adventure geeft een probleem, opdracht of vraag. De speler geeft een commando of tegenvraag. En het programma reageert weer op dit commando of deze opdracht. In feite kun je elk spel, waarbij je op een toets moet drukken of een spelpookje aanwendt interactief noemen. Wij maken echter een duidelijk onderscheid tussen de adventures die volgens een vast protocol opereren en diegene waar een creatieve inbreng van de speler wordt gevraagd. Het neerschieten van een vliegende schotel is heel wat anders dan het zorgvuldig plannen van een coup of het opzetten van een valstrik voor de achtervolgde misdadiger.

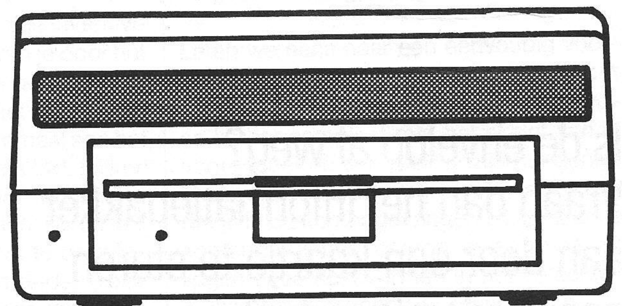
Laten we eens naar een eenvoudig voorbeeld kijken: Die akelige dwerg was je net te vlug af en je zit nu vast in een onderaardse kerker compleet met het skelet van je voorganger en een paar dozijn hongerige knaagdieren. Het programma registreert dat je in de kerker gevangen zit en gaat de komende acties daarop afstemmen. Er liggen twee actievormen in het verschiet:
1. De speler probeert te ontsnappen.
2. De ratten beramen een feestmaal.
En misschien komt die kobold ook nog terug.
Stel, de ratten doen een poging en de speler blijkt gewapend met een (onderweg vergaarde) sabel. Het adventure valt in het voordeel van de speler uit en er is geen leven verspeeld. Nu heeft de speler tijd om aan zijn ontsnapping te werken. De deur ziet er stevig uit, evenals de muren. Maar wellicht kun je het slot open krijgen of is er een slecht afgesloten luchtschacht. Een schrandere adventurist heeft zich eerder op zijn tocht misschien van explosieven voorzien en blaast de deur uit zijn scharnieren.

In een uitgekiend adventure kan een dergelijke actie op zich nieuwe gebeurtenissen uitlokken. Bijvoorbeeld de terugkeer van de (nu woedende en zwaar bewapende) dwerg of het instorten van de kerkers. Een goed interactief adventure speelt in op de vondsten van de spelers en laat hen binnen de spelregels tal van vrijheden. Desondanks zijn er alternatieve wegen om het avontuur tot een goed einde te brengen.

De videobeeldplaat zal in de nabije toekomst het begrip interactief er een dimensie bijgeven. Namelijk die van de al hiervoor besproken "graphics"- en soundrealiteit waardoor de spelers zich zelf sterk met het adventure gaan identificeren. ●

Iedere gebruiker van een Commodore C-64 met een 1541-diskdrive kent de problemen van de trage overdracht van programma's van de computer naar de diskdrive en omgekeerd. Dat de traagheid het gevolg is van de seriële poort van de C-64, is al in vele artikelen beschreven. De C-64 beschikt naast deze poort ook over de expansie-poort, die de signalen parallel verwerkt en tevens over een aantal stuursignalen beschikt. Maar op deze poort kan de 1541-diskdrive niet worden aangesloten. In de tijd van steeds goedkoper wordende hardware is het aantrekkelijk, om een "kale" diskdrive van een ander merk aan te schaffen en deze op de expansie-poort van de C-64 aan te sluiten. Met behulp van een interface en een Disk Operating System zijn fascinerende resultaten te bereiken. Dit artikel van de heer van Putten geeft de ervaringen weer van de opbouw en het werken met zo'n snelle diskdrive en de verdere mogelijkheden.

Snelle Diskdrives als hobby



Het werken met een diskdrive heeft een aantal voordelen boven lezen/schrijven naar een cassette recorder, omdat de data-overdracht tussen de computer en de diskdrive sneller verloopt en de data op de diskette rechtstreeks is te benaderen. Bij de cassette recorder worden de data altijd achter elkaar op de cassette tape (sequentieel) geschreven en is het noodzakelijk, om eerst door te spoelen of te zoeken naar het begin van een bepaald bestand of programma.

Een ander voordeel van de diskdrive is het wijzigen van al eerder weggeschreven data. De ruimte voor de gewijzigde data is normaliter aanwezig, terwijl het op de cassette tape niet mogelijk is om een bestaand programma uit te breiden, zonder de data van een volgend programma te overschrijven. Uiteraard kan een kortere versie wel zonder problemen op tape worden geschreven, maar daarbij ontstaan nutteloze tussenruimtes.
Het gebruik van een Commodore C-64 met een 1541-diskdrive heeft wel het voordeel van de rechtstreekse benadering van de data, maar is relatief traag in de overdracht van de data. Dit laatste is het gevolg van de seriële overdracht van de data, omdat de afzonderlijke bits van een byte achterelkaar worden overgestuurd. Een aanzienlijke winst in de overdrachtssnelheid wordt bij het parallel oversturen van de bits bereikt. Als een byte bijvoorbeeld uit 8 bits bestaat, is het parallel oversturen 8 maal sneller dan serieel. De parallelle mogelijkheid is beschikbaar op de expansie-poort van de C-64. Bovendien zijn op deze poort de stuursignalen aanwezig, voor het aansturen van een diskdrive.
In de elektronica hobby wereld worden momenteel diskdrives aangeboden tegen aantrekkelijke prijzen. Voor een bedrag onder de 500 gulden is al een dubbelzijdige diskdrive te koop. Het probleem is echter, dat het een zogenaamde kale diskdrive betreft, dus zonder kast en voeding en die niet rechtstreeks op de expansie-poort is aan te sluiten. Een passende interface en een geschikt Disk Operating System (DOS) zijn daarnaast, of moeilijk te verkrijgen, of de aanschafprijs ligt behoorlijk hoog. Met deze achtergrond wordt er eerst een wensenlijst opgesteld, waaraan de diskdrive moet voldoen:

- De overdrachtssnelheid van de data moet groter zijn dan van de 1541-diskdrive.
- De C-64 mag intern niet worden gewijzigd door bijvoorbeeld andere ROM's te plaatsen.
- Het Disk Operating System mag het direkt beschikbare geheugen in de C-64 niet gebruiken.
- Het Disk Oprating System mag niet meer dan 8 K-bytes geheugen in beslag nemen.
- Het aanroepen van de disk kommando's moet rechtstreeks geschieden, dus net zoals de Basic-kommando's.
- De totale aanschafkosten mogen niet boven de prijs van een gebruikelijke 1541-diskdrive uitkomen.

Zo te zien stelt deze lijst een behoorlijk aantal voorwaarden. Na heel wat afwegingen en onderzoek resulteert de lijst in:

- het kopen van een diskdrive
- het bouwen van een buffer voor de expansie - poort
- het bouwen van een interface tussen de C-64 en de diskdrive
- het schrijven van een Disk Operating System.

Deze komponenten zullen achtereenvolgens nader worden bekeken.

De Diskdrive

Voor de diskdrive wordt het merk TEAC gekozen, met als type FD-55B. Deze diskdrive werkt met dubbelzijdige 5-1/4 inch diskettes, heeft een redelijk korte accesstijd en beschikt over een zogeheten Shu-

gart-kompatibele input/output konnektor. Bovendien ligt de prijs momenteel onder de 500 gulden. Daar komt nog wel de prijs van een passende metalen kast en een voeding bij. De totale opslagkapaciteit is afhankelijk van de opslagstruktuur op de diskette en bedraagt ongeformateerd 250 K-bytes bij FM-modulatie en 500 K-bytes bij MFM-modulatie. De modulaatvorm bepaalt de wijze, waarop de data op de diskette wordt geschreven. In het eerste geval worden synchronisatiepulsen tussen de data-bits geschreven, terwijl in het tweede geval deze pulsen ontbreken. Door het ontbreken van de pulsen kan de schrijfdichtheid op de diskette, en daardoor overdrachtssnelheid twee maal zo hoog worden. Maar dit stelt hoge eisen aan de signalen, die door de interface moeten worden verwerkt. Daarom is er gekozen voor FM- modulatie. In dit geval is de overdrachtssnelheid tussen de C-64 en de diskdrive 125.000 bits per seconde. Bij een byte-lengte van 10 bits, namelijk een startbit 8 databits en een stopbit, komt dit overeen met 12,5 K-bytes per seconde.

De Buffer

Het doel van de buffer is om de expansiepoort te beveiligen tegen overbelasting en kortsluiting en de signalen tussen de C-64 en de interface te verzorgen. Hiertoe zijn de adreslijnen en de datalijnen voorzien van (bidirektionele) buffers en wordt de klokpuls uit de microprocessor van de C-64 vertraagd. Dit laatste is nodig, omdat de timing van de microprocessor-signalen, zoals deze in de C-64 gebruikt wordt, niet direkt geschikt is voor de interface. Het geheel is opgebouwd op een dubbelzijdige elektronica-print en ondergebracht in een kunststof kastje, dat rechtstreeks in de expansie-poort wordt geplaatst. De voedingsspanning wordt direkt uit de expansie-poort betrokken. De andere zijde van de print is via een printkonnektor en een stuk bandkabel met de interface verbonden.

De Interface

Dit is de belangrijke schakel tussen de C-64 en de diskdrive, omdat deze de elektrische signalen op de juiste wijze verzorgt en doorgeeft. De interface bevat naast een paar TTL-IC's een zogeheten PIA (Peripheral Interface Adapter), die de sturing van de poorten van de diskdrive regelt. Tevens bevat de interface een ACIA (Asynchronous Communications Interface Adapter), die de data-overdracht tussen de computer en de diskdrive verzorgt. De ACIA zet het door de C-64 aangeboden parallelle data-byte om in seriële data-bits voor de diskdrive en omgekeerd. De snelheid bedraagt 12,5 K-bytes per seconde. Als laatste bevat de interface een Eprom, die 8 K-bytes groot is en waarin het Disk Operating System (DOS) is opgeslagen. Hiermee wordt de DOS geheel buiten de C-64 gehouden en is door de disk-kommando's direkt te benaderen.
De interface kan 4 enkelzijdige -, of 2 dubbelzijdige diskdrives selekteren en aansturen. In dit ontwerp wordt voorlopig 1 dubbelzijdige diskdrive gebruikt. Ter beperking van de kosten is de interface op een enkelzijdige elektronica-print gebouwd en in de metalen kast van de diskdrive ondergebracht. De voedingsspanningen voor de interface en de diskdrive worden door een aparte voeding, buiten deze kast verzorgd.

Het Disk Operating System

Het schrijven van het Disk Operating System (DOS) heeft verreweg de meeste tijd gekost. Eerst moet namelijk de indeling op diskette worden bepaald. Omdat dezelfde opbouw voor de 1541- diskdrive, door het varierende aantal sectoren per spoor haast onmogelijk is, kunen de bestaande diskettes voor de 1541- diskdrive niet gebruikt worden, of de interface vergt een forse uitbreiding. Na diverse afwegingen wordt er gekozen voor een andere indeling op de diskette. In vergelijking met de diskettes voor de 1541-diskdrive bevat het eerste spoor nu de inhoudsopgave (directory) en geeft het al of niet bezet zijn van de sectoren op de overige sporen aan. Een sector of data-blok is 256 bytes groot en per spoor worden 8 sectoren geschreven. De beide zijden van de diskette bevatten elk 40 sporen, zodat een diskette meer dan 1160 K-bytes kan opslaan.
De DOS is ontwikkeld met behulp van de PAL-Assembler. Dit programma zet de, in een Basic-programma geschreven assemblertaal om naar machinekodes. Het voordeel hiervan is, dat wijzigingen in de assemblertaal eenvoudig zijn aan te brengen en er tevens een gedokumenteerd Basic-programma ontstaat. De DOS kommando's met een schrijfopdracht naar de diskette voeren altijd een leesopdracht uit. Hierdoor vindt er een kontrole op het schrijven plaats.
De DOS is in vier delen te splitsen, namelijk:
- de initialisatie
- de besturing
- de subroutines
- de kommando's

De initialisatie van de DOS dient om de IC's van de interface te resetten en de DOS te aktiveren. In de tweede regel op het beeldscherm verschijnt nu "64K DOS SYSTEM" in plaats van "64K RAM SYSTEM". Dit geeft het aktief zijn van de DOS aan. De DOS kommando's kunnen nu net zoals de gebruikelijke Basic-kommando's worden ingetoetst. Het verschil is, dat de DOS kommando's worden voorafgegaan door het /-teken.
De besturing van de DOS kontroleert of het ingegeven DOS kommando bestaat en laat vervolgens naar het startpunt van dit kommando springen. Bij een onjuist kommando of onrechtmatigheden tijdens het uitvoeren van een kommando wordt door de besturing de foutafhandeling geregeld. Andere taken van de besturing zijn het aktiveren van de DOS-ROM op de interface-print en het onderdrukken van de Basic- Interrupts. Door dit laatste wordt een snelheidswinst tijdens het uitvoeren van het DOS kommando bereikt, omdat nu niet iedere keer wordt gekeken of er een interrupt is gegeven. Als het DOS kommando is uitgevoerd, schakelt de besturing de DOS-ROM af en geeft het gebruikelijke Basic van de C-64 weer vrij.
De subroutines zijn programma-delen die door verschillende DOS kommando's worden gebruikt. Als voorbeelden zijn te noemen:
- het starten of stoppen van diskdrive
- het kontrole op write - protect
- het zoeken van een spoor en een sector
- het lezen of schrijven van een sector
- het zoeken van een programma-naam in de inhoudsopgave

De DOS kommando's zijn in twee groepen te verdelen. De eerste groep bevat de basis kommando's om met de diskdrive te kunnen werken met de diskdrive aantrekkelijker maken, of als utilities (hulpprogramma's) zijn te beschouwen.

De basis kommando's zijn:
- Format
- Load
- Save
- Delete
- Replace
- Restore
- Kill
- Stop

De extra kommando's zijn:
- Copy
- Verify
- Rename
- Memory
- Block
- Record

Met uitzondering van het Replace, Rename, Restore en Stopkommando worden de kommando's afgekort tot de eerste letter. Zoals bijvoorbeel /L"TESTPROG" voor het laden van een programma.
Hierdoor zijn de afkortingsmogelijkheden van de C-64 vereenvoudigd, omdat het gebruik van de Shift-toets vervalt.
Met de basis kommando's zijn dus de elementaire handelingen met een disk-

drive uit te voeren. Ze worden in het kort behandeld.

Het **FORMAT** kommando dient om eventuele oude data op een diskette te wissen en de sectormarkers te plaatsen. Tevens worden de sectoren voor de inhoudsopgave (directory) opgezet. De sectormarkers zijn per spoor genummerd, waarmee de DOS kontroleert of de juiste sector wordt geraadpleegd. Ieder spoor wordt direkt gekontroleerd op de bereikbaarheid van de sectormarkers. Indien na twee leespogingen een sectormarker niet is gevonden, wordt het formateren onderbroken en verschijnt er een foutmelding.

Het **LOAD** kommando kan drie taken uitvoeren, namelijk het laden van een Basic -of machinetaalprogramma van de diskette in de computer, het kontroleren van een weggeschreven programma, of het opvragen van de inhoudsopgave. Tijdens het laden van een programma wordt voor elke sector de data-pariteit berekend en vergeleken met de sector-pariteit. De kontrolerende taak van het load kommando dient om de sectoren van de inhoudsopgave en van een weggeschreven programma direkt te kontroleren op eventuele fouten. Deze kontrole is een onderdeel van het Save Kommando.

Bij het opvragen van de inhoudsopgave wordt het beeldscherm eerst gewist en toont daarna de inhoud van telkens twee sectoren. Dit komt overeen met maximaal 16 programma's per beeldscherm. Hierdoor is een zogenaamde doorblader mogelijkheid aanwezig en wordt de inhoudsopgave niet achterelkaar doorgeschoven. Omdat de inhoudsopgave rechtstreeks op het beeldscherm verschijnt, wordt een eventueel aanwezig Basic-programma niet overschreven.

Het **SAVE** kommando kontroleert eerst of de diskette beschreven mag worden (write-protected is) en voldoende vrije sectoren bevat om het programma weg te kunnen schrijven. Het opvullen van de sectoren geschiedt vanaf het laagste spoornummer, zodat de lege sectoren, na bijvoorbeeld een Delete Kommando, direkt weer gebruikt worden. Dit geeft een optimale sector bezetting, maar leidt bij veelvuldig verwijderen van programma's tot een inefficient gebruik van de diskette. Want het laden van zo'n "gesplitst" programma duurt langer. In dit geval zal het kopiëren van de diskette de enige oplossing zijn, om de sectoren weer bij elkaar te krijgen.

Het SAVE kommando kan zowel Basic- als machinetaalprogramma's wegschrijven. Bij machinetaalprogramma's is het verplicht, om het begin- en eindadres op te geven.

Na het schrijven van de sectoren wordt direkt de schrijf-kontrole uit het LOAD kommando uitgevoerd. Als hierbij geen fouten zijn gekonstateerd, worden de sectoren als bezet gemarkeerd en als laatste de inhoudsopgave bijgewerkt.

Het **DELETE** kommando kontroleert eerst of de diskette write- protected is en geeft vervolgens de gebruikte sectoren weer vrij. De programma-naam in de inhoudsopgave wordt daarna voor normaal gebruik onbereikbaar gemaakt. De inhoudsopgave zelf wordt verder niet verstoord, zodat een eventueel uitgewist programma is te herstellen, tenzij de betreffende sectoren weer voor een ander programma zijn gebruikt.

Het **REPLACE** kommando kontroleert eerst of de diskette write- protected is en vervangt vervolgens de oude versie van een programma op de diskette door een nieuwe. Het Replace kommando is naar verhouding eenvoudig, want het is een kombinatie van het Delete- en het Save kommando.

Het **RESTORE** kommando maakt een in de computer aanwezig Basic- programma weer beschikbaar. Dit kan bijvoorbeeld nodig zijn na het resetten van de interface, of het abusievelijk intoetsen van het New kommando.

Het **KILL** kommando schakelt het gebruik van de DOS kommando's uit en zet de Basic-Sectoren terug naar de oorspronkelijke waarden van de C-64.

Het **STOP** kommando biedt de mogelijkheid, om het loopwerk van de diskdrive uit te schakelen. Dit is bijvoorbeeld nodig na het abrupt onderbreken van een DOS kommando.

De snelheid

De resultaten met een snelle diskdrive zijn zonder meer fascinerend te noemen. Maar het zal duidelijk zijn, dat de daadwerkelijke ontwikkeling niet probleemloos is verlopen. De voorbereidingen, de opbouw en het oplossen van diverse problemen heeft, als vrijetijdsbesteding ongeveer een jaar in beslag genomen, terwijl op dit moment de extra DOS kommando's nog in ontwikkeling zijn. Dit laatste is onder andere het gevolg van de geheugen grootte van de C-64, want het Basic-programma met de assemblertaal beslaat al een gebied van bijna 30 K-bytes. Daarom zal dit Basic-programma in twee gedeeltes moeten worden ontwikkeld. Een voordeel is het hierbij behorende machinetaal- programma, want dit past met zijn 4 K-bytes in het $C000 geheugengebied van de C-64. Hierdoor kan het gebruik van de DOS-Eprom uitgesteld worden en zijn eventuele wijzigingen in de DOS gemakkelijk aan te brengen.

Om een indruk te geven van de snelle verwerking, volgt hier aan aantal cijfers. Het formateren van een diskette neemt ongeveer 50 seconden in beslag. Het laden van een programma van 4 K-bytes duurt iets meer dan 2 sec. en het laden van 32 K-bytes gebeurt binnen 8 seconden. Voor het starten van de diskdrive en het op snelheid komen van de drivemotor is ongeveer 1 seconde nodig. Vandaar dat het laden van een programma van 32 K-bytes, naar verhouding sneller verloopt. Het wegschrijven van een programma duurt bijna 2x zolang als het laden van dit programma, omdat elke schrijfopdracht direkt wordt gevolgd door een schrijf-kontrole. Deze kontrole is te vergelijken met het laden van het programma.

In het kader van dit artikel zou het te ver voeren, om de gedetailleerde schema's van de elektronica-printen of de komplete DOS af te drukken.

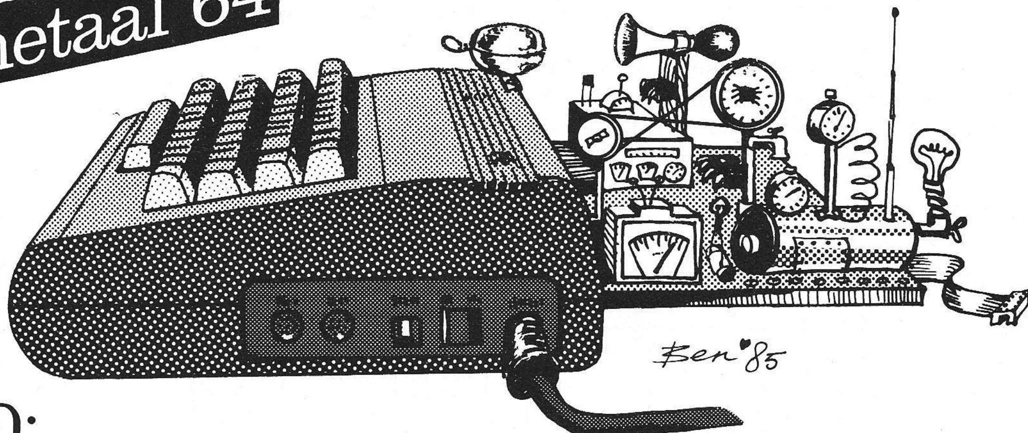
Wel kan het gebruik ervan heel wat plezier opleveren, want het werken met zo'n diskdrive is net alsof je met een racewagen tussen de oldtimers rijdt. En het blijft nog een hobby ook. ●



De 128 heeft met de 1570/71 al een iets snellere diskdrive.

Deze aflevering gaat over de achtergronden van machinetaal en de manier waarop de computer communiceert. Bij de Commodore's zijn er standaard-routines voor in- en uitvoer. Die z.g.n. KERNAL-routines zijn speciaal voor gebruik vanuit machinetaal ontwikkeld. Ze zijn op alle Commodore-computers aanwezig.

Serie Machinetaal 64



Programmeren in machinetaal

Door Sjoerd Bakker

Aflevering 10:

De KERNAL's I/O-routines.

Zoals bekend is het op de Commodore 64 mogelijk om in Basic te programmeren. Dit lijkt zo op het eerste gezicht een vanzelfsprekend feit, maar dat is het niet. Een computer is in wezen niets meer dan een apparaat, dat uitsluitend onderscheid kan maken tussen een lage en een hoge spanningen. Doordat hij dit onderscheid echter op honderdduizend verschillende plaatsen in de computer kan maken, en omdat die spanning wel honderdduizende keren per seconde veranderd kan worden, wordt een computer pas redelijk bruikbaar.

Het is wel nodig om een zekere orde te scheppen in die "spanningsplaatsen" en de verandering ervan, alvorens we met de computer aan de slag kunnen. Een mens communiceert namelijk niet in termen van lage en hoge spanning/stroom, maar (o.a.) met symbolen, die we letters en cijfers noemen. Die orde is in de computer op verschillende niveau's aanwezig. Met een laag niveau bedoelen we in dit verband dan het computerniveau en met een hoog niveau het mensenniveau.

Niveaus

De orde op het laagste niveau, de microcode, treffen we aan in het binnenste van de microprocessor. Daar zijn duizenden (transistor)schakelingen zodanig geschakeld, dat ze op bepaalde invoer naar de registers reageren met afgeronde bewerkingen, zoals het optellen van twee getallen. Dit is het domein van de Flip-Flops, de Acid-Trips, en de NOT en NAND-poorten.

Het tweede niveau is bekend terrein voor de lezers van deze rubriek. Het is het machinecode-, oftewel het machinetaalniveau. Losse machinetaalopdrachten hebben, net als individuele schakelingen, in het algemeen weinig nut. Pas als je verschillende opdrachten samenvoegt, kan er een zinvol geheel ontstaan, zoals bijvoorbeeld een routine, die een letter op het scherm zet.

Het derde niveau is het niveau van de "hogere" programmeertaal, zoals Basic. Ook hier geldt weer dat een enkele opdracht weinig zin heeft, en dat er dus weer een geheel van opdrachten, een programma, nodig is om iets tot stand te brengen. Samenvattend kunnen we zeggen, dat een Basic-programma is samengesteld uit een aantal opdrachten; en zo'n Basic-opdracht wordt dan omgezet (door de interpreter) tot een serie machinetaal-opdrachten. Iedere machinetaal-opdracht veroorzaakt dan weer een serie schakelingen in de micro-processor. Overigens is het allemaal niet zo duidelijk afgebakend als ik het hier voorstel, maar dit "pyramide-principe" is in elk computersysteem aanwezig.

ASCII-CODES

Door de hele computer heen speelt nog een andere ordening een rol. Omdat de mens, zoals gezegd, communiceert d.m.v. letters en cijfers, kunnen we met maar één plaats van hoge of lage spanning weinig beginnen. We zouden er hoogstens het verschil tussen een A en een B mee kunnen symboliseren. Maar als we twee van die plaatsen tegelijk zouden kunnen beschouwen en met de combinatie van de twee spanningsniveaus zouden kunnen werken, dan zou het mogelijk zijn om vier letters voor te stellen. Bijvoorbeeld: laag/laag=A, laag/hoog=B, hoog/laag=C en hoog/hoog=D. Ook dat is lang niet genoeg, dus heeft men uiteindelijk gekozen voor groepjes van 8 bij elkaar horende informatiedragende schakelingen. Hiermee hebben we (2x2x2x2x2x2x2x2=) 256 verschillende mogelijkheden om met laag-hoog etc. een teken voor te stellen. Lage en hoge spanning stellen we trouwens voor met 0 en 1, die eenheden van informatie noemen we bits (binary digits), en een groepje van 8 bij elkaar horende bits noemen we een byte of geheugenplaats. Data wordt dus telkens met 8 bits tegelijk verplaatst. Dat geldt met betrekking tot de processor, chips en geheugenplaatsen. Via het tweetallig stelsel kunnen de 0/1-kombinaties vertalen naar decimale of hexa-decimale waarden. We kunnen met

een byte dus de waarden $00-$FF voorstellen. Nu kunnen we die waarden natuurlijk op verschillende wijzen laten corresponderen met bijvoorbeeld letters. Dat is een kwestie van afspraken maken. Verreweg de meeste computers en randapparatuur werken volgens de ASCII-afspraak (American Standard Code for Information Interchange).
Bijgaand een lijst met karakters en hun ASCII-code, voor zover deze voor de Commodore 64 van belang zijn. De meest linkse kolom geeft voor een karakter de

ASCII-CODES

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0														cr		
1																
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	

waarde van de Lo-nybble en de bovenste rij geeft de waarde van de Hi-nybble. De codes, kleiner dan $20 zijn besturingstekens. Zoals je ziet, gebruikt Commodore er maar ´één van; het besturingsteken CR (code $0D) of Carriage Return heeft als gevolg, dat de "cursor" naar het begin van de volgende regel gaat (return-toets).
Het is nuttig om te weten dat de ASCII-code officieel een 7-bits code is. Bit 7 wordt normaal gesproken gebruikt voor pariteitscontrole. Maar het is de laatste tijd gebruikelijk geworden om ook codes boven de $7F te gebruiken voor bijvoorbeeld een cursieve karakterset. Daarom heb ik op eigen gezag bit 7 voor alle codes 0 gemaakt. Tot zover dit uitstapje naar de ASCII-codes.

ROM's

Je kunt dus in Basic programmeren dankzij het feit dat de '64 beschikt over een aantal standaard machinetaal-routines. Deze routines zitten in twee ROM's gebakken: de Basic-ROM ($A000-$BFFF) en de KERNAL-ROM ($E000-$FFFF). Grofweg gezegd, bevinden de rekenroutines zich in Basic-ROM en de I/O-routines in de KERNAL-ROM. Deze routines zijn vanuit eigen machinetaal-programma's normaal te gebruiken met een JSR. Een aantal van de KERNAL-routines heeft echter van Commodore een speciale status gekregen. Vanaf lokatie $FF81 wordt voor hen in alle Commodore-computers op vaste plaatsen 3 bytes gereserveerd. Deze drie bytes bevatten een JMP-instructie naar het begin van de eigenlijke routine, dat dan op alle computers verschillend kan zijn.

Een voorbeeld: JSR $FF9F heeft als gevolg een JMP naar $EAB7, waardoor het toetsenbord wordt gescand en het A-register de ASCII-code van de gedrukte toets bevat. Die JMP kan dus in elke computer naar een andere plaats gebeuren, maar de JSR naar de $FF9F moet altijd het lezen van het toetsenbord als gevolg hebben.

I/O-Routines

Na deze lange inleiding komen we toe aan het eigenlijke onderwerp van deze aflevering. We zullen de KERNAL-routines gaan bekijken, die betrekking hebben op de in- en uitvoer van de Commodore-computers. Per routine wordt vermeld met welke registers je ze moet aanroepen, en wat het resultaat is van de routine.

OVERZICHT KERNAL-ROUTINES

ADRES	JMP VIA	NAAR	SOORT	NAAM	OMSCHRIJVING
$FF81		$FF5B	ALG	CINT	INITIALISEER SCREENEDITOR
$FF84		$FDA3	ALG	IOINIT	INITIALISEER I/O-CHIPS
$FF87		$FD50	ALG	RAMTAS	TEST RAM
$FF8A		$FD15	ALG	RESTOR	INITIALISEER I/O-VEKTOREN
$FF8D		$FD1A	ALG	VECTOR	LEES/INIT. I/O-VEK. GEBRUIKER
$FF90		$FE18	ALG	SETMSG	PRINT ERROR/CONTROL MELDINGEN
$FF93		$EDB9	SER	SECOND	VERSTUUR SEC. ADRES NA LISTEN
$FF96		$EDC7	SER	TKSA	VERSTUUR SEC. ADRES NA TALK
$FF99		$FE25	ALG	MEMTOP	LEES/ZET TOP GEHEUGEN
$FF9C		$FE34	ALG	MEMBOT	LEES/ZET BODEM GEHEUGEN
$FF9F		$EA87	KEY	SCNKEY	ONTVANG KARAKTER VAN KEYBOARD
$FFA2		$FE21	SER	SETTMO	ZET TIME OUT
$FFA5		$EE13	SER	ACPTR	ONTVANG KARAKTER VAN SERIEBUS
$FFA8		$EDDD	SER	CIOUT	VERSTUUR KARAKTER NAAR SERIEBUS
$FFAB		$EDEF	SER	UNTLK	VERSTUUR UNTALK
$FFAE		$EDFE	SER	UNLSN	VERSTUUR UNLISTEN
$FFB1		$ED0C	SER	LISTEN	VERSTUUR LISTEN
$FFB4		$ED09	SER	TALK	VERSTUUR TALK
$FFB7		$FE07	I/O	READST	LEES STATUS
$FFBA		$FE00	I/O	SETLFS	ZET FILE-PARAMETERS
$FFBD		$FDF9	I/O	SETNAM	ZET FILE-NAAM
$FFC0	($031A)	$F34A	I/O	OPEN	OPEN FILE
$FFC3	($031C)	$F291	I/O	CLOSE	CLOSE FILE
$FFC6	($031E)	$F20E	I/O	CHKIN	BEPAAL INVOER-APPARAAT
$FFC9	($0320)	$F250	I/O	CHKOUT	BEPAAL UITVOER-APPARAAT
$FFCC	($0322)	$F333	I/O	CLRCHN	I/O APPARATEN ALS NORMAAL
$FFCF	($0324)	$F157	I/O	CHRIN	ONTVANG KARAKTER
$FFD2	($0326)	$F1CA	I/O	CHROUT	VERSTUUR KARAKTER
$FFD5		$F49E	I/O	LOAD	LAAD (PROGRAMMA-) BLOK
$FFD8		$F5DD	I/O	SAVE	SAVE (PROGRAMMA-) BLOK
$FFDB		$F6E4	ALG	SETTIM	ZET KLOK
$FFDE		$F6DD	ALG	RDTIM	LEES KLOK
$FFE1	($0328)	$F6ED	KEY	STOP	TEST STOP-TOETS
$FFE4	($032A)	$F13E	I/O	GETIN	ONTVANG KARAKTER
$FFE7	($032C)	$F32F	I/O	CLALL	CLOSE ALLE FILES
$FFEA		$E69B	ALG	UDTIM	VERHOOG KLOK
$FFED		$E505	ALG	SCREEN	LEES SCHERMFORMAAT
$FFF0		$E50A	ALG	PLOT	LEES/ZET CURSOR-POSITIE
$FFF3		$E500	ALG	IOBASE	LEES BASISADRES I/O-CHIP

VERKLARING TABEL SOORT:

ALG = ALGEMENE ROUTINE
I/O = ALGEMENE I/O-ROUTINE
KEY = KEYBOARD-ROUTINE
SER = SERIEBUS-ROUTINE

Bijgaande tabel bevat achtereenvolgens voor alle routines het KERNAL-adres, eventueel een indirekte JMP, een afkorting betreffende de soort routine, de afkorting van de routine zoals door Commodore gebruikt, en tenslotte een korte omschrijving van de routine. Sommige JMP's zijn indirekt via twee RAM-geheugenplaatsen. Het adres, dat daarna wordt vermeld, is de waarde, die deze geheugenplaatsen normaal gesproken bevatten. Als gebruiker kun je deze vektoren echter laten wijzen naar het begin van je eigen routine als het geknutsel van de Microsoft-proppers je niet bevalt.

1. SETLFS (JSR $FFBA)
A-register = File nummer
X-register = Apparaat nummer
Y-register = Secundair Adres (opdrachtcode)
Deze routine zet de parameters klaar voor de OPEN-routine. De gebruikte nummers corresponderen met die van de OPEN-opdracht in Basic. Als bit 7 van de inhoud van het Y-register 1 is, wordt er geen Secundair Adres verstuurd.

2. SETNAM (JSR $FFBD)
A-register = Lengte File naam
X-register = Lo-byte adres File naam
Y-register = Hi-byte adres File naam

SETNAM zet de lengte van de File-naam en het adres vanaf waar deze in het geheugen staat, klaar voor de OPEN-routine. Door het A-register 0 te maken, geef je te kennen niet met een File-naam te willen werken. Het X- en Y-register hoeft dan niet te worden ingevuld. Het werken met File-naam heeft alleen zin bij Cassette- of Disk-I/O. In het geval van RS-232 communicatie wijst de X/Y-pointer niet naar een File-naam, maar naar de eerste van 4 bytes die de baudrate bepalen. Zie de Reference Guide.

3. OPEN (JSR $FFC0)

OPEN kan zonder ingevulde registers worden aangeroepen. Wel moeten van te voren de routines SETLFS en SETNAM zijn uitgevoerd (mag ook in omgekeerde volgorde). OPEN voegt de geopende File toe aan de lijst met andere open Files. Er kunnen maximaal 10 Files tegelijk geopend zijn. Voor de functie van de secundaire adressen verwijs ik weer naar de Reference Guide. De KERNAL-routines SETLFS, SETNAM en OPEN vervullen samen de funktie van de Basic-opdracht OPEN.

4a. CHKIN (JSR $FFC6)

X-register = File nummer

Met CHKIN bepaal je welk apparaat zal dienen als invoer-toestel. De invoer-routines halen dan van dit apparaat hun karakters. Onder karakters versta ik trouwens in I/O verband een ontvangen of verstuurde byte-waarde. Het X-register moet het File-nummer bevatten. CHKIN zoekt dan zelf in de tabel het bijbehorend apparaat op. Normaal gesproken zal de invoer van het toetsenbord komen.

4b. CHKOUT (JSR $FFC9)

X-register = File-nummer

CHKOUT werkt hetzelfde als CHKIN, alleen wordt nu het uitvoer-toestel gekozen. Het resultaat kun je vergelijken met de CMD-opdracht in Basic. Voor CHKIN en CHKOUT geldt, dat de bewuste File eerst geopend moet zijn.

5a. GETIN (JSR $FFE4)

Voor het gebruik van deze routine hoeven geen registers te worden ingevuld. Na het aanroepen bevat het A-register een karakter van het met CHKIN gedefinieerde apparaat. Bij invoer vanaf het toetsenbord (0) zal een karakter uit de keyboard-buffer worden gehaald en bij invoer vanaf het scherm (3) wordt het karakter onder de cursor in het A-register geladen. De karakters zijn de ASCII-waarden en anders Commodore's eigen codes. Zie de R.G. Een waarde van $00 betekent: geen toets ingedrukt.

5b. CHRIN (JSR FFCF)

Ook CHRIN valt zonder ingevulde registers aan te roepen. Deze routine haalt, net als de vorige, een karakter op van het invoer-toestel naar het A-register. Ook verder is deze routine hetzelfde als GETIN, behalve bij invoer van het toetsenbord (0) en een RS 232-apparaat (2).

In het eerste geval wordt een speciale procedure gevolgd. De invoer vindt helemaal niet plaats vanaf het toetsenbord, maar vanaf het scherm. Het gaat als volgt: ingetypte karakters worden zonder knipperende cursor naar het scherm uitgevoerd tot er een return volgt. Dan wordt het eerst ingetypte karakter van het scherm gehaald, in het A-register gezet, en er volgt een RTS. Bij elke volgende aanroep wordt weer een volgend ingetypt karakter van het scherm gehaald. Dit kan zo doorgaan, tot we alle karakters hebben gehad. Wordt de routine dan nog een keer aangeroepen, dan zal het A-register de waarde 0D (re-

turn-code) bevatten, ten teken dat we klaar zijn.
CHRIN van het toetsenbord valt dus te vergelijken met de Basic-opdracht INPUT. Het nadeel van deze werkwijze is, dat je er niet mee kunt vaststellen of er bijvoorbeeld een funktietoets is ingedrukt. Deze laten geen sporen op het scherm achter. In dat geval is dus de GETIN-routine beter op zijn plaats. Het grote voordeel is, dat je een afgerond stuk informatie kunt inlezen, zonder je te hoeven bemoeien met "gedelete" tekens etc.

5c. CHROUT (JSR $FFD2)

A-register = Te versturen karakter-code.
CHROUT is de tegenhanger van CHRIN. Je kunt er een karakter mee zenden naar het met CHROUT gespecificeerde apparaat. De uitvoer geschiedt, behalve naar het beeldscherm, altijd gebufferd. Naar RS 232 en Cassette wordt er eerst gewacht tot er een blok is volgeschreven, en vervolgens wordt dit blok in zijn totaliteit verstuurd. Bij uitvoer naar de seriebus bedraagt de buffering precies 1 karakter. Het laatste karakter wordt door UNLSN verstuurd. CHROUT kun je vergelijken met de Basic-opdracht PRINT.

6. CLRCHN (JSR $FFCC)

Deze routine beëindigt de in- en uitvoer met de huidige apparaten. De invoer komt vervolgens weer vanaf het toetsenbord, en de uitvoer gaat weer naar het scherm. Het aanroepen van de routine heeft in het geval van de seriebus UNLSN (inclusief verzenden van het laatste karakter), dan wel UNTLK als gevolg. CLRCHN dient dus te worden aangeroepen als de communicatie met het betreffende apparaat (voorlopig) beëindigd moet worden.

7. CLOSE (JSR $FFC3)

A-register = File-nummer
Door de CLOSE-routine wordt het betreffende File-nummer uit de administratie van de computer geschrapt. Hierna houdt de File op te bestaan en kun je er geen karakters meer naar sturen of van ontvangen. Closen, doe je een file dus pas, als je hem nergens in je programma meer gebruikt.

Dat waren dan de 7 stappen voor communicatie met wat voor apparaat dan ook. Er komt per apparaat natuurlijk nog heel wat meer bij kijken, dus heb je voor een "gespecialiseerd" gebruik van de I/O-mogelijkheden wel een wat meer gedetailleerde kennis van de verschillende routines nodig. Daarvoor zijn er twee bekende boeken op de markt, die de volledige ROM-listing van de '64 bevatten, voorzien van uitleg en commentaar. Het zijn "64 Intern" van Angerhausen etc. en "Commodore 64 binnenste buiten" van B. de Jong. Helaas kan ik ze geen van beiden van harte aanbevelen.

"64 Intern" (Duitse versie) stopt soms bladzijden lang met commentaar (bijvoorbeeld bij de Garbage Collection), en bevat weliswaar niet overdreven veel, maar toch nog te veel fouten. In het boek van De Jong staat per bladzijde commentaar gemiddeld minstens één fout of onvolledigheid en dat is een heel voorzichtige schatting. Daar staat dan tegenover, dat het commentaar van "64 Intern" heel begrijpelijk is en het boek een schat aan informatie over de '64 biedt, en dat De Jong zeer veel commentaar geeft en op sommige gebieden echt goed is (o.a. Cassette-I/O). Kortom: als globale leidraad zijn deze boeken zeker bruikbaar, maar voor de juiste details zul je vaak je eigen gezonde verstand moeten gebruiken.

LOAD/SAVE

Dan gaan we verder met de behandeling van nog wat "losse" routines, te weten LOAD, SAVE, CALL en READST. Deze hebben wel met de algemene I/O te maken, ze passen niet in het 7-stappen schema van daarnet.

LOAD (JSR $FFD5)

A-register = LOAD (0) of VERIFY (< > 0)
(X-register = Lo-byte laad-adres)
(Y-register = Hi-byte laad-adres)
Voorafgaand aan de LOAD-routine moeten SETLFS en SETNAM zijn aangeroepen. De LOAD-routine kent een belangrijke extra faciliteit ten opzichte van de Basic LOAD-opdracht. Dit geldt voor zowel Cassette als Disk. De inhoud van het A-register bepaalt of er geload (0) dan wel geverified (ongelijk 0) wordt.
Het secundair adres van de SETLFS-routine speelt een belangrijke rol. Is dit ongelijk 0, dan wordt volgens het Basic-principe LOAD "FILE-NAAM", 1,1 of 8,1 geladen. Maar is het secundair adres 0, dan wordt het programma geladen naar vanaf het adres, dat de X- en Y-registers bij aanroep bevatten. Er wordt dus in dat geval niet zoals in Basic zonder meer geladen naar het begin van de Basic-RAM ($0801). Om dit in bovenstaand geval wel te bewerkstelligen, moet je dus het X-register met $01 en het Y-register met $08 vullen bij de LOAD-aanroep. Een andere manier om te loaden is met secundair adres ongelijk 0. Elk programma bevat immers het adres vanaf waar het gesaved is en dat geldt ook voor Basic-programma's.

SAVE (JSR $FFDB)

A-register = Lo-byte adres pointer begin blok
X-register = Lo-byte adres +1 eind blok
Y-register = Hi-byte adres eind blok
Voorafgaand aan SAVE moeten weer de routines SETLFS en SETNAM zijn aangeroepen. De routine zorgt voor het saven van een blok data waarvan het beginadres gevormd wordt door een pointer in Zeropage. Het A-register moet bij aanroepen de Lo-byte van het adres van deze pointer bevatten, en de X- en Y-registers samen het eindadres+1 van het blok.
Een voorbeeld: je wilt een Basic-programma saven, waarvan de start dus altijd $0801 is. Je zet de waarden $01 in lokatie $FB en de waarde $08 in de lokatie $FC. Vervolgens laadt je het A-register met de waarde $FB en de X- en Y-registers met het eindadres+1 van het programma. Dan roep je de routine SAVE aan.

CLALL (JSR $FFE7)

Door het aanroepen van deze routine worden alle open Files geclosed. Zie verder CLOSE.

READST (JSR $FEB7)

Na aanroepen van deze routine bevat het A-register de z.g.n. Status-waarde. Dit is de inhoud van een adres in het O.S.-RAM waarvan de bits elk een eigen informatiewaarde hebben. Bijgaand overzicht geeft aan, welke betekenis aan het 1 zijn van zo'n bit gehecht moet worden.
Daarmee zijn we weer aan het eind gekomen van deze aflevering. Op de andere KERNAL-routines zullen we als het zo uitkomt nog wel eens terugkomen, maar ze zijn verder te specifiek gericht om er nog een hele aflevering aan te besteden. Niet iedereen heeft bijvoorbeeld een apparaat aan de seriebus hangen, terwijl 9 van de 39 KERNAL-routines wel aan de seriele I/O zijn gewijd.

Tot de volgende keer.

BETEKENIS GEZETTE STATUS-BITS

BIT	CASSETTE READ	SERIAL I/O	CAS. LOAD/VERIFY
0		TIME OUT WRITE	
1		TIME OUT READ	
2	SHORT BLOCK		SHORT BLOCK
3	LONG BLOCK		LONG BLOCK
4	READ ERROR		LOAD/VERIFY ERROR
5	CHECKSUM ERROR		CHECKSUM ERROR
6	END OF FILE	E O I	
7	END OF TAPE	DEVICE NOT PRESENT	END OF TAPE

Tekstverwerken op de C64/C128-machines is momenteel geen enkel probleem meer. Het aanbod aan wordprocessors is groot en een oude Easy Script-versie is al voor minder dan 40 gulden te koop. Natuurlijk laten de goedkopere tekstverwerkers voor de fervente "wordsmith" nog een aantal wensen open. Het prijsverschil met de "professionele" tekstpakketten zoals Superscript en Vizawrite is echter tamelijk groot, zodat men die paar nadelen gauw voor lief neemt. De nieuwe Personal Word Processor "HOMEWORD" van Aackosoft, oorspronkelijk van Amerikaanse "Sierra" bodem, brengt hier wellicht verandering in, omdat de prijs en de prestaties een goed compromis vormen.

HOMEWORD

Een Ikoon-tekstverwerker met allure

Een tekstverwerker met een prijskaartje van 175 gulden moet meer in zijn mars hebben dan de reeks tekstpakketten, die voor minder dan ƒ 100,- in het schap van de computershop liggen. Het eerste dat je opvalt aan plastic boek/doos verpakking, is dat het om een ikoongestuurde tekstverwerker gaat. Homeword biedt de verwende thuiscomputeraar, scholen en de kleine zakenman veel (semi-)professionele faciliteiten voor een vriendelijke prijs, scoort goed op vrijwel alle punten zoals printer-ondersteuning en teken-aanpassing, maar is door de diskwisselingen wat traag in het gebruik.

Homeword werkt met ikonen en dat is niet erg gebruikelijk. Het gebruik van ikonen ofwel beeldsymbooltjes voor functies, wordt niet door iedereen als optimaal ervaren. Aan de positieve zijde staan het gebruiksgemak en de overzichtelijkheid. Er zijn echter vele computergebruikers, die toch liever met functietoetsen en gewone menu's werken of de opdrachten gewoon willen intypen. Een ander punt is dat de ikoon-graphics natuurlijk wat extra beslag op het RAM-geheugen leggen, waardoor er minder programma- en tekstruimte overblijft. Bij Homeword heeft men het programma verdeeld over een aantal modules, zodat de tekstruimte vrij groot (meer dan 14 vel A-4) is, maar dat maakt het werken ermee wat moeilijker. Voor printen en andere functies moet men namelijk van schijf wisselen.

Documentatie

Een Nederlandstalige handleiding is een vrij zeldzaam genoegen bij serieuze C-64 software. Aackosoft, de importeur van dit Sierra pakket en dus verantwoordelijk voor de handleiding heeft zich goed van haar taak gekweten. Zelden zagen wij zo'n duidelijke en overzichtelijke opzet in een Nederlandse vertaling.

De eerste 12 van de 70 pagina's vormen een soort rondleiding door de verschillende mogelijkheden van HomeWord in verhaalvorm. De eigenlijke handleiding bestaat uit overzichtelijke hoofdstukjes waarin de mogelijkheden kort, helder en doelmatig behandeld worden.

Verder bevat de handleiding een groot aantal voorbeelden, een paar "ikonen"-stambomen (waarin de ikoonmenu's worden uitgelegd), een verklarende woordenlijst en een index met trefwoorden. Als geheugensteuntje is er een referentiekaart met de meest gebruikte functies/toetsen. Zijn er toch aanmerkingen op de handleiding? Ja en nee. Door de opzet om ook voor de totale computerleek begrijpelijk te zijn, kan de tekst voor de gevorderde computeraar wellicht enigzins ergelijk overkomen. In de handleiding komt ook niet duidelijk genoeg naar voren dat je eerst een nieuwe disk dient te formatteren. Na het desbetreffende diskutility moet je HomeWord namelijk opnieuw inladen en daarbij gaat de reeds ingetypte tekst verloren. Voordat je gaat formatteren waarschuwt het programma zelf wél voor het tekstverlies. Daar schiet je echter weinig mee op als er geen beschrijfbare schijf voorhanden is.

Het laden

HomeWord is alleen op disk leverbaar en u ontkomt dan ook niet aan het gebruik van een of liefst twee drives. Die tweede drive is wel nuttig, omdat het programma voor wat betreft bepaalde onderdelen in afzonderlijke blokken geladen wordt. Als je bijvoorbeeld van het EDIT-menu naar de DISK-utilities wilt, zal het programma eerst het desbetreffende blok van de schijf moeten laden. Met slechts één drive is dat dus programma-filedisk e.o. wisselen geblazen. Bij veel korte tekstjes is Homeword dus arbeidsintensief en traag. Overigens geen echt gebruiksprobleem als je de opeenvolgende handelingen een beetje logisch opzet.

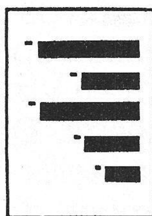
Het laden zelf gaat via **LOAD"HW",8** (Let er wel op dat deze tweezijdige disk aan de andere kant een ATARI-versie bevat!). Type **RUN** en daarna wordt je tijdens de laadtijd van ruim twee minuten getrakteerd op wat reklameteksten van SIERRA. Na een laatste tekstscherm doorlopen te hebben kom je bij het hoofdmenu, dat is opgebouwd uit ikonen.

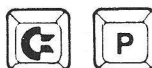
Functies

HomeWord werkt vrijwel geheel ikonen-

gestuurd. De ikonen staan onder een soort pointerlijn onderin het beeld.
Het verplaatsen van het ikonenvenster, een vierkantje dat je om de actief te maken ikoon moet zetten, gaat met behulp van de cursor- en SHIFT-toetsen en de functie wordt actief na het indrukken van de RETURN-toets.
In de startsituatie staat het keuzevierkantje om het **Layout**-ikoon. Verder biedt het hoofdmenu een **Printer-**, een **Edit** (tekst wissen, verplaatsen, dupliceren, woord opzoeken), een **File-**, een **Customize-** en een **Diskutility-** ikoon.
Via de **F1**-toets kom je bij het normale invoerscherm voor tekst. Deze F1-toets is erg belangrijk en wordt ook gebruikt om van de diverse submenu's weer naar het hoofdmenu terug te komen of een opdracht ongedaan te maken.
De diverse edit/layout-functies kun je direkt vanuit het hoofdmenu activeren. Maar de Print-, File-, Customize- en Diskutility-programmablokken worden na activatie van het desbetreffende ikoon eerst van de HomeWord-disk geladen. Dit is een compromis tussen het gebruikersgemak en het vrijhouden van het geheugen voor tekst, maar wel ten koste van wat snelheid.

Indented point





inspringende punten <OPTION> en <P> *(Commodore toets en <P>)*

Print-opties

Via de Print-functie kun je de papiersoort (tractor/los), de paginanummering, de viewfunctie en Print Document instellen. Het aantal instelmogelijkheden is dus vrij beperkt, maar gelukkig is via het speciale Customize menu wel degelijk een hele reeks printers ondersteund en kunnen bijzondere lettertekens worden ingesteld (accenten).

Edit

In de redactie-mode, dus via het Edit-ikoon zijn er diverse mogelijkheden. Via **Erase Text**, het ikoon toont een pagina met een pijltje naar het vuilnisbakje, kun je teksten wissen met de cursor. Te ver gegaan? Gewoon de cursor terug laten keren. Een nieuwe RETURN wist de geselecteerde tekst. Als je de gewiste regels later elders weer in het document wilt gebruiken moet je het ikoon **Insert Erased Text** (zelfde plaatje, maar met een omgekeerd pijltje) gebruiken. Verder bevat dit submenu nog de funkties **Move Text, Copy Text, Find** en **Find and Replace** (correctie van een woord door de gehele tekst).

File en Layout

Het File-ikoon biedt de opties **Insert** (invoegen), **Get** (pak/load), **Save** en **Include** (merge).
Met behulp van het Layout-ikoon zijn de regelafstand (Set Spacing, drie onderlinge afstanden en kantlijnen), regelindeling (Alignment), inspringende punten (Indented Point), lettertype (Print Style) en titels (Headings/Footings) instelbaar. Met **Start New Page** ga je naar de volgende pagina. Bij de lettertypen kun je uit vet, normaal en

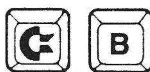
onderstrepen kiezen.
Customize is een belangrijk ikoon voor de bezitters van niet-Commodore printers, meerdere diskdrives en voor hen die de vooraf ingestelde margins willen wijzigen. Alle wijzigingen kun je, na het verwijderen van het schrijf-protectiestickertje, op de HomeWord-disk vastleggen. Verder kun je met **Make Backup Documents** een file extra saven.
Tot slot het **Disk Utility** menu. Na het laden van dit programmablok verschijnen de ikonen voor Prepare Document (formattingopdracht), Erase Document (scratch), Catalog (index) en EXIT(uit). Verder is er ook nog een huisje waarmee je gegevens van en naar andere programma's van de Home-serie kunt brengen. Dat is de manier, waarop men samen kan werken met bv. een database-pakket.

Het tekstscherm

Voor het intypen van de tekst is er het "normale" tekstscherm. Het onderste derde deel van het scherm wordt door een horizontale lijn begrenst. Daaronder staan links het paginanummer, de hoeveelheid vrij RAM-tekstgeheugen (in grafische vorm), de nog beschikbare fileruimte op de

Boldface text

H



druk tekst vet af <OPTION> en <B>
(Commodore toets en <B>)

disk (eveneens een grafiek) en de aanwijzing "Press F1 to go to the menu". Rechts staat een vertikale lijn, waarnaast een grafische voorstelling van de paginaindeling en -vulling te zien is om een goed overzicht te krijgen.
Boven de lijn kun je de naar boven scrollende tekst typen, er is in beeld ruimte voor maximaal 15 regels. Dat is vrij beperkt, maar het overzichtje helpt je bij de orientatie. Met de cursortoetsen loop je door de tekst. De DELETE-toets wist de gemaakte typefouten. De spatiebalk geeft spaties en insertruimte. M.b.v de CONTROL-toets wandel je voor-/achteruit door de verschillende pagina's. De F3-toets regelt de Tabs(tabulatorstanden) en de RETURN-toets beëindigt een paragraaf of voegt (twee maal drukken) een extra regel toe. Verder kun je ook gewoon over een foutieve tekst heen typen.
De speciale layout/print-commando's verschijnen in reversed video blokjes in de schermtekst.

Homeword is een ikoongestuurde tekstverwerker met professionele allures.

Binnen de beperkingen van de C64/C128 biedt dit Aackosoft programma veel waar voor zijn geld. Het is even wennen aan al die Ikonen en de vertraging die door het laden van de verschillende blokken optreedt. Eenmaal ingeburgerd beschik je over een veelzijdige wordprocessor die thuis, op school of op een klein kantoor niet zal misstaan en ook een brede range printers ondersteunt. ●

SOFT WIR WAR

DOOR BERT TIER.

